LOW POWER DIGITAL INTEGRATED CIRCUIT DESIGN FOR AN UNUSUAL EVENT DETECTOR

Robert Ervin Bruninga

United States
Naval Postgraduate School

THESIS

LOW POWER DIGITAL INTEGRATED CIRCUIT DESIGN FOR AN UNUSUAL EVENT DETECTOR

by

Robert Ervin Bruninga

Thesis Advisor: H.A. Titus

June 1971

Approved for public release; distribution unlimited.

Low Power Digital Integrated Circuit Design For an Unusual Event Detector

by

Robert Ervin Bruninga
Ensign, United States Navy
B.E.E., Georgia Institute of Technology, 1970

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
June 1971

ABSTRACT

A deep ocean capsule is proposed capable of sensing pressure, temperature, flow and direction for a study of tides and internal waves of the oceans. One of the major requirements for this capsule is the detection and recording of unusual events. To meet this need a very low power, highly reliable digital detector has been designed. COS/MOS logic with maximum use of MSI and LSI was used in the design. The detector was simulated using proposed word lengths and fixed point arithmetic which would be utilized in the ensuing hardware. The fixed point simulation was necessary to verify the problems in truncation and round-off errors and to provide necessary design criteria to ensure proper filter functioning and integral detection.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

I. INTRODUCTION

An unusual event detector is to be a part of a one-year deep ocean experiment [Ref. 1]. The sensors and recording system are all digital and have low power and high reliability as overriding design constraints.

The detector is to have several distinct subsystems as shown in Figure 1: a spurious data rejection gating system; a semi-diurnal and diurnal tide rejection filter; and a square and integrate fading memory detector. Upon determination of an unusual event characterized by significant energy in the spectrum from 0.25 CPH to 30 CPH [Ref. 1], various permanent memory and accoustic telemetry links may be initiated. The sensor signals are to be sampled at one-minute intervals and stored in a two-hour buffer memory. If no event is occurring, the signals are lost at the end of two hours. If an event is occurring, the buffer memory is tied into the permanent memory.

The spurious data rejection system is accomplished by tapping the buffer memory at the half-hour points and taking first differences of adjacent signals. If the differences exceed a certain tolerance level, a spurious data point or glitch is assumed, and the first difference is rejected and is replaced with an appropriate value. These five first differences are then combined with appropriate weighting to achieve the anti-tide filtering. The resulting signal is

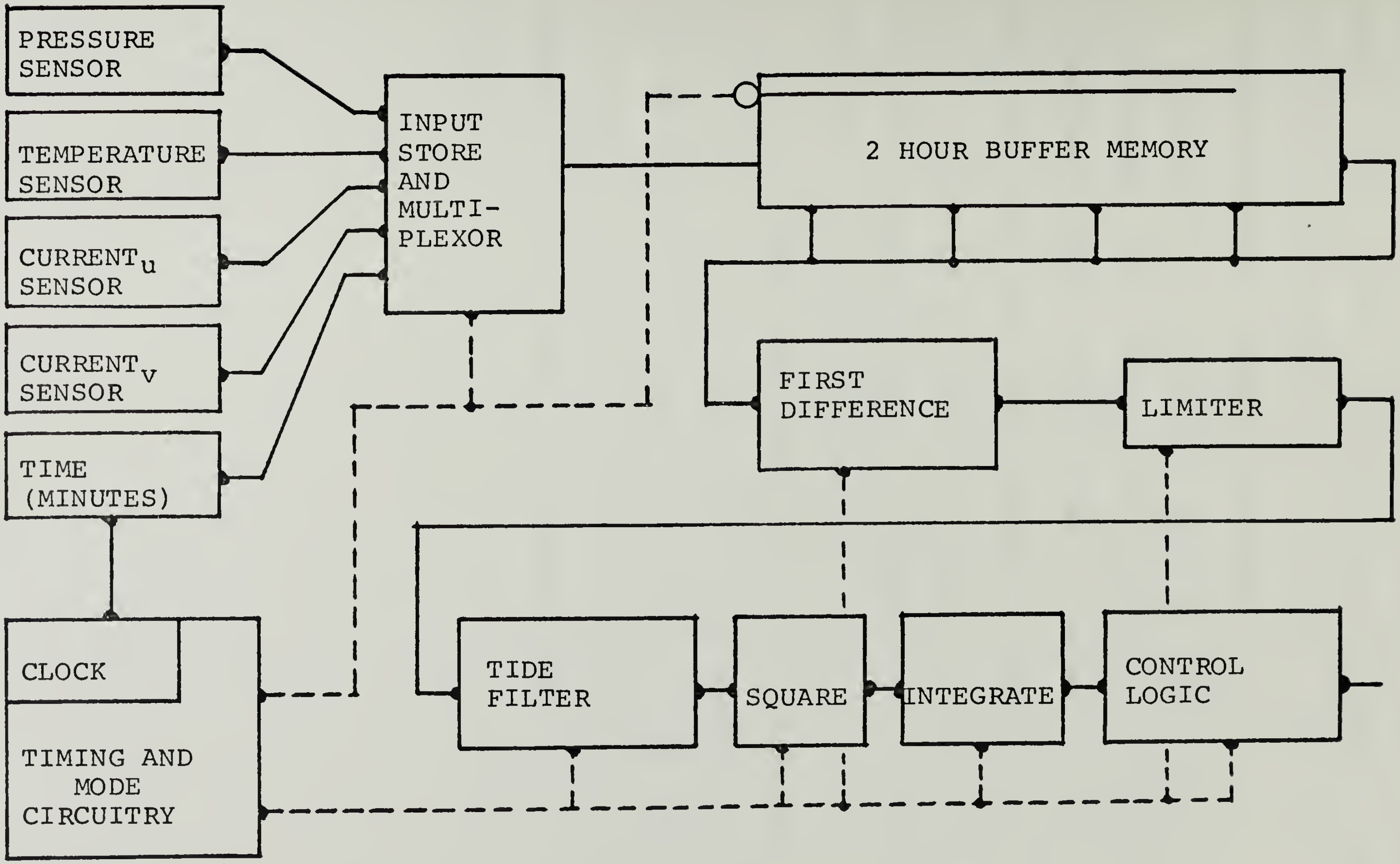


Figure 1. Unusual Event Detector - Pressure Input is Monitored for an Unusual Event Which Will Trip Control Circuitry to Activate Various Devices

then squared and fed into an integrator with a fading memory time constant of one-half hour. That is, the detector will be predominantly sensitive to events that have accumulated over the past half-hour.

Because the system is to operate unattended at the 3,000-fathom level for one year, power and reliability are tantamount design considerations. For this reason, COS/MOS IC's are to be used with maximum utilization of existing LSI and MSI components. Based on these considerations, the following hardware design is proposed as a preliminary prototype for system evaluation.

A logic design was formulated, and its veracity was analysed via a simulation using actual sensor word lengths and appropriate fixed point arithmetic. The simulation as a design tool verified an optimum design with proper trade-offs between system performance and power and size projections.

II. DESIGN CONSIDERATIONS

A. SYSTEM EQUATIONS

Although the four input variables of temperature, pressure, and the two current components are sampled, only the pressure data is sampled for unusual events in the tsunami spectrum of 1 to 30 CPH [Ref. 1]. A more sophisticated device could use any number of detectors on any of the inputs using the same temporary storage buffer memory.

1. First Difference and Limiter

In order to remove the mean pressure and long-term drift, the first stage in the detector is the first differencer. It computes the difference between successive samples $P_t - P_{t-1}$. In order to protect the detector from any spurious data which would give a false indication, a limiter which limits the first difference according to the following relation is the next step in the detector:

$$\delta P_t = \begin{cases} P_t - P_{t-1} & \text{if } \delta P_t \leq 3\sigma \\ \delta P_{t-1} & \text{if } \delta P_t > 3\sigma \end{cases}$$

2. Tide Filter

The tide filter is a fourth order filter with zeroes chosen to remove the diurnal and semidiurnal tides. Two sections of the form

$$\delta_t + a_1\delta_{t-\Delta} + b_1\delta_{t-2\Delta} = \int \delta(t-\tau)W_1(\tau)d\tau \quad \text{where}$$

$$W_1(\tau) = 1 + a_1\delta(\Delta) + b_1\delta(2\Delta)$$

with a_1, b_1 and a_2, b_2 chosen to give the required zeroes are combined by convolution to give the fourth order difference equation:

$$W(z) = 1 + a_1a_2\delta(\Delta) + \left[a_1a_2 + b_1b_2\right]\delta(2\Delta) + \left[a_1b_2 + a_2b_1\right]\delta(3\Delta) + b_1b_2\delta(4\Delta)$$

With $\Delta = 1/2$ hour, the five inputs to the filter will be the first difference δP_t at the times 0, -30, -60, -90, and -120 minutes. The constants are calculated to be

$$\begin{aligned} a_0 &= 1 \\ a_1 + a_2 &= -3.920392 \\ a_1a_2 + b_1b_2 &= 5.841800 \\ a_1b_2 + a_2b_1 &= -3.920392 \\ b_1b_2 &= 1 \end{aligned}$$

3. Integration

To obtain the power spectral density of the tide filter the signal is next squared and averaged over a fading function with a 60 minute time constant.

$$\overline{P^2}(t) = a\overline{P^2}(t-\Delta) + P^2(t) \qquad a \lesssim .991666$$

The only function remaining is to determine if the power has exceeded a preset threshold indicating significant energy in the tsunami band to signal an "event."

4. Control Functions

The system proposal calls for a dump of the previous two hours of one-minute samples whenever a tsunami is

detected. This will ensure that an adequate history leading up to an event is stored. Also included in the logic is the circuitry to continue storing the one-minute samples until the second consecutive two-hour interval which is below threshold. Extra logic gates are provided, however, for a variety of controls lending flexibility to the detector as a "black-box" component.

B. SYSTEM ARCHITECTURE

The logic design for a special purpose digital machine begins with the system equations, block diagrams and a general understanding of overall system requirements. Special purpose structures can usefully be described by the general system pictured in Figure 2 illustrating the two major parts of a special purpose machine, the data processing and timing and mode circuitry.

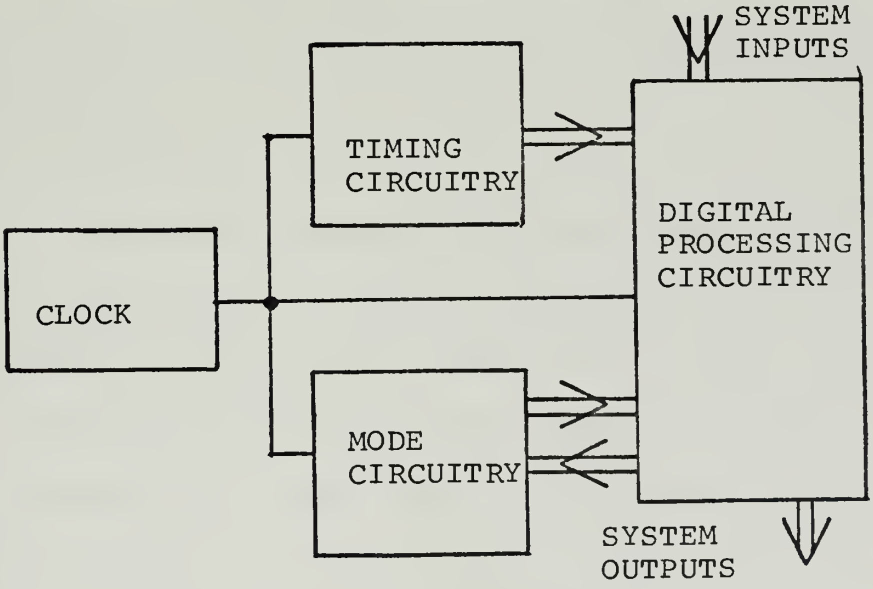


Figure 2. Special Purpose Structure

The data processing circuitry includes all input and output interfacing circuits, the data operators and system blocks, any storage of memory devices, and, in general, the necessary hardware to take system inputs and derive system outputs. The timing and mode circuitry serves much the same purpose for the special purpose structure as the software program does for the general purpose computer. The timing circuitry sequences operations in a set pattern from some initial time. The mode circuitry controls operations which are functions of the data being handled as well as time.

In considering the best approach to the design of the digital package, the need for low power has repeatedly been stressed. For this reason, serial processing circuitry which has the advantage of economy of hardware at the expense of speed is used predominantly with only slight modifications to meet other of the specified system requirements.

1. Memory

The major component and power requirement of the detector will be the temporary data storage section or buffer memory. Investigation into this problem centered around core memories, shift registers, thin films and delay lines. Core memories have the definite advantages of nonvolatility, simplicity, zero quiescent power, reliability and random access capability. The same is basically true of the new thin film adaptations of ferrite storage. Shift registers

are popular as short term buffer storage, are serial in operation, and require no external read write hardware.

Since random access, nonvolatility of information, and access time are of little importance here, the most favorable devices are long shift registers which are compatible with the logic family. A discrete state shift register of length 10,000 bits at first seems to be rather prohibitive, but recent trends in MSI are making these long shift registers well-suited for small size memories. Using current low power MOS technology as many as 64 static bits are available on a single chip. Since such long shift registers used as a recirculation memory in a serial machine need simply a single port input, a single port output and a clock, the total amount of hardware involved for the complete memory is very small.

Also considered were dynamic shift registers, which have more than three times the number of bits per chip as static registers [Ref. 2]. Although the quiescent power is lower, the total power is greater than for static devices because of the transient power of continuous clocking. The resulting long static discrete state shift register chosen, therefore, was dictated by available hardware at low power, serial operation for simplicity, recirculation at an extremely low rate for reduced hardware count and data accessibility, and low clock rate for low duty-cycle. The organization of data in the shift register memory is shown in Figure 3.

2. Data Organization

The type of data organization and corresponding arithmetic code to be used for the detector results from the form of the input data. All five input channels (pressure, temperature, current components and time) will be sixteen bit binary magnitudes. For this reason, the Data can be simply word organized within the samples which are sequentially organized as shown in Figure 3. The arithmetic code chosen was initially a one's complement code for simplicity. Further study, however, indicated the use of a two's complement code was more compatible with serial processing and would ultimately require fewer operations and hardware.

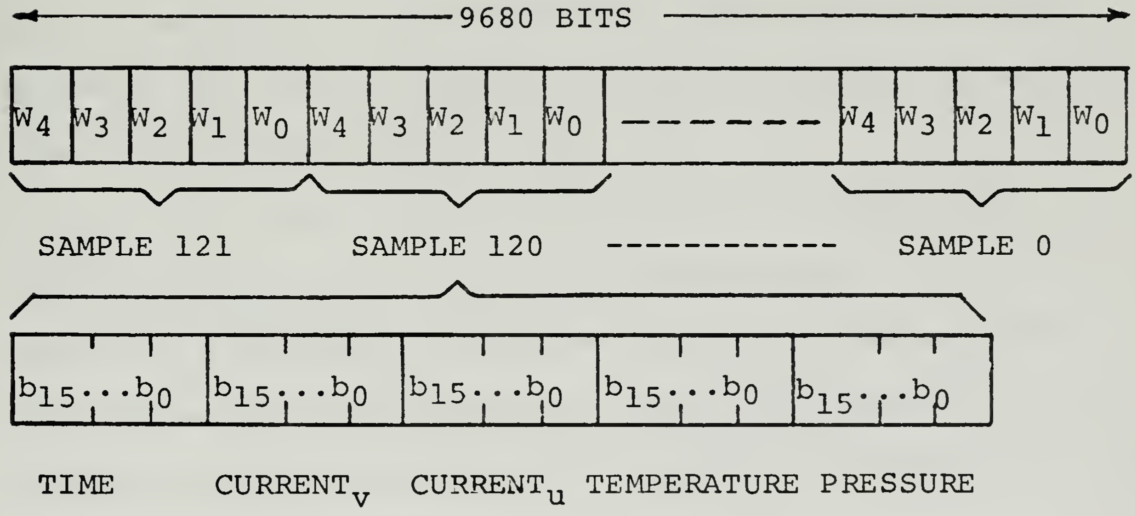


Figure 3. Data Organization in Memory

Two additional considerations which will be clarified in the later section regarding the use of RCA's COS/MOS logic family are the use of positive AND/OR logic for initial logic design and the use of asychronus timing circuits. The short-cuts involving the use and marriage of NAND and NOR logic to reduce the package count is a matter of circuit juggling to make optimum use of the available gating packages. Asychronus or pseudosynchronus timing of all operations again results from the tradeoff of speed for simplicity.

3. Flexibility

Two of the specifications for the unusual event detector are compatibility and flexibility. The input section of the device is therefore designed to accept a variety of input combinations. The digital sensors for the deep sea in most cases are frequency modulated signals corresponding to the particular input variable. They are measured by allowing the output frequency to count a binary counter for some predetermined period [Ref. 3].

The input section is designed to accept five variable frequency or variable period signals. The exact period of measurement (from one second to four minutes) or the choice between frequency or time measurement is easily changed, using removable solder jumpers to program the input in a variety of fashions.

The storage of the 120 minute samples in a static register provides a flexibility of output. A simple

clocking gate or even an external clock can be used to read out this information in almost any format.

C. IMPLEMENTATION WITH COS/MOS

Recent trends in semiconductor technology toward medium and large scale integration have made the MOS transistor increasingly attractive due to its ease of manufacture and versatility of operation. Single chip building blocks of rather large arrays, registers, and memories are being incorporated into many logic families; of primary interest here, however, is the use of the MOS device in a complementary symmetry arrangement which has the distinct advantage of extremely low power [Ref. 4].

At present the leading supplier of a complete logic line using complementary symmetry MOS transistors is Radio Corporation of America with its COS/MOS line. The COS/MOS CD4000A series of integrated circuits features a variety of options including special low voltage types, high reliability circuits, various packaging styles, and the usual military or industrial temperature ranges.

1. Complementary Symmetry Circuitry

The breakthrough that made complementary symmetry logic practical in medium scale integration was the ability to fabricate opposite polarity, but similar devices in large numbers on the same chip. The basic gate circuit from which all other logic functions are derived is shown in Figure 4. The usual load resistor is replaced with another FET of opposite polarity; thus, there is no quiescent current

except for the leakage currents on the order of nanoamps. The pair of MOS transistors are simply switches which tie the output to either the supply voltage or to ground. In true complementary logic, the circuit for the P devices is the dual network function of the N units.

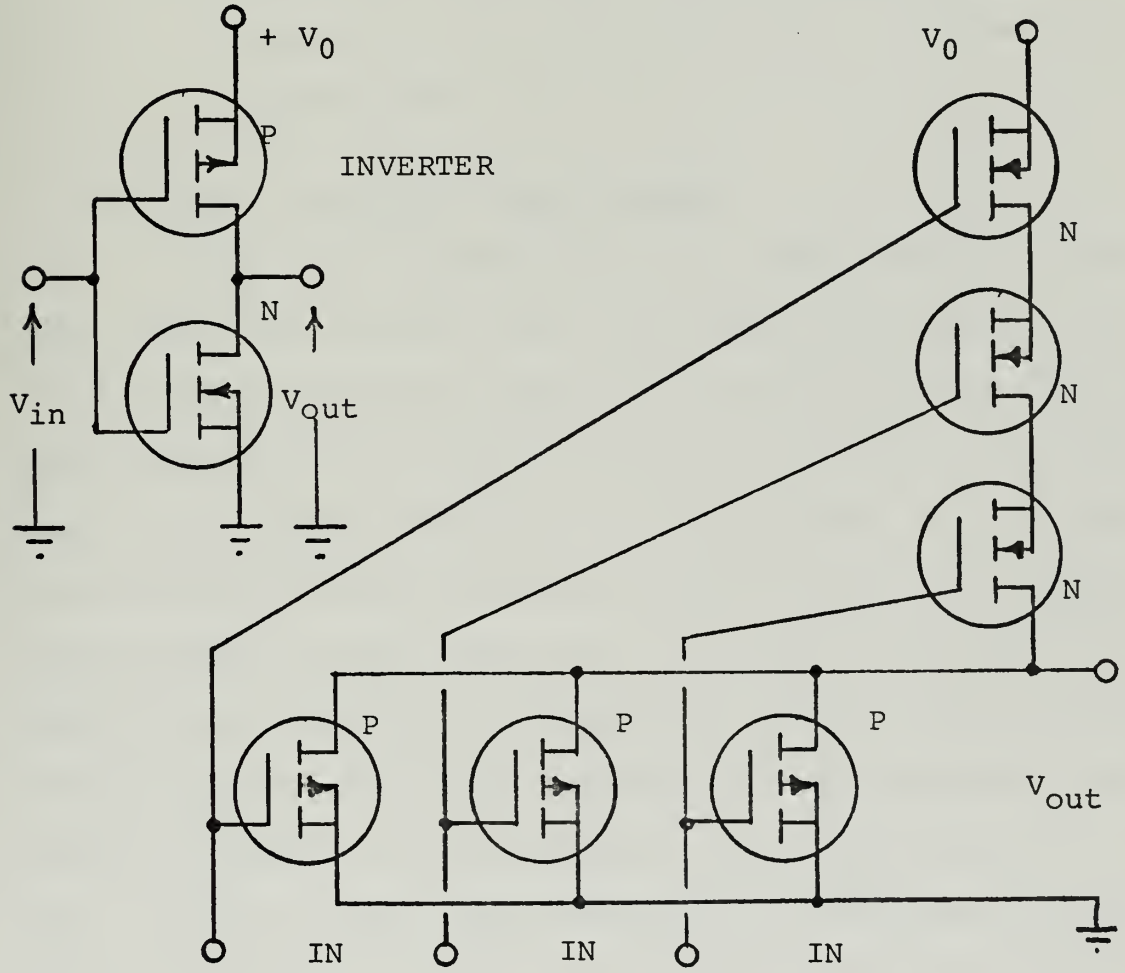


Figure 4. Three Input Complementary Symmetry MOS NOR GATE

The advantages derived by the complementary arrangement are more than just the microwatt standby power:

a. Simple one supply direct-coupled logic using only MOS transistors

b. Large fanout capability because of capacitive inputs

c. Good noise immunity due to large threshold voltages

d. High speed operation because output node capacitance is always charging or discharging through the "on" unit

2. Shift Registers Using COS/MOS

The P and N channel units can form a shift register using true complementary logic with two flip-flops per bit for two phase operation; however, a significant reduction in components can be obtained by the use of (1) the excellent capacitance storage feature of the MOS gates, and (2) the two halves of the MOS flip-flop, one of which will provide the function of delay [Ref. 4]. As shown in Figure 5, opening the switches S_1 and S_2 which form the cross-coupling connections results in two separate storage elements, each capable of storing for an extended but finite period of time. In this state, new information can be input to the left gate, while the level of the right half is transferred to the next stage on the right. Closing the switch S_2 and then closing S_1 locks the one-zero information indefinitely. The two phase clocking requires a rather quick transfer of data from the left to the right halves of a stage, but an indefinite delay is allowed between stages.

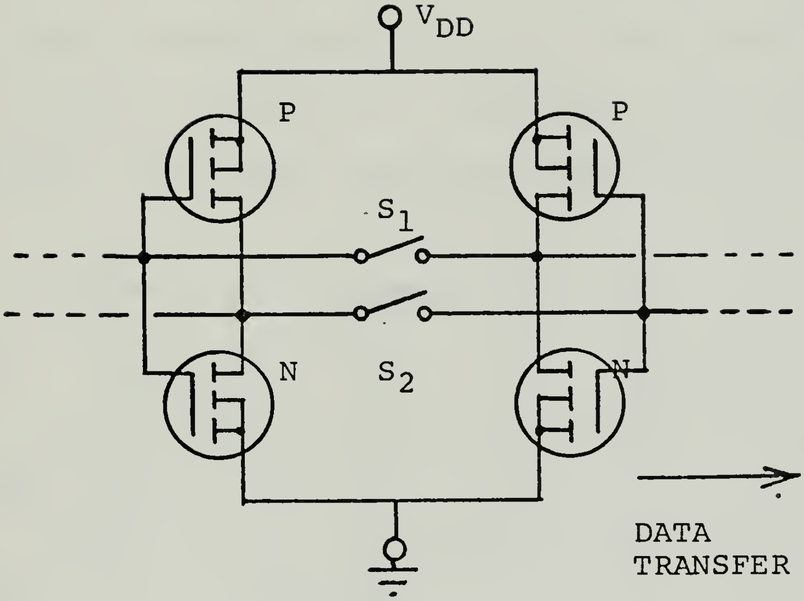


Figure 5. Basic Shift Register Stage

3. Power Dissipation

Considering the total power dissipation as the sum of both the quiescent and transient powers, further power reduction can be obtained by proper design of the switching characteristics. Minimizing the logic level swings, using true complementary loads, and minimizing leakage currents as well as all output and load capacitances form the general approach to an optimum system. The total power P dissipated in a complementary set-reset flip-flop is given by

$$P = 2C_0V_0^2f + P_s$$

where C_0 is the node capacitance, V_0 is the supply voltage, f is the operating frequency, and P_s is the standby power.

For low switching power, the supply voltage V_0 must be reduced to the lowest possible value. At a V_0 of -4 volts, and an output capacitance C_0 of 23 pf the frequency dependence is shown in Figure 6. Note, also, the large increase in quiescent power with temperature [Ref. 4].

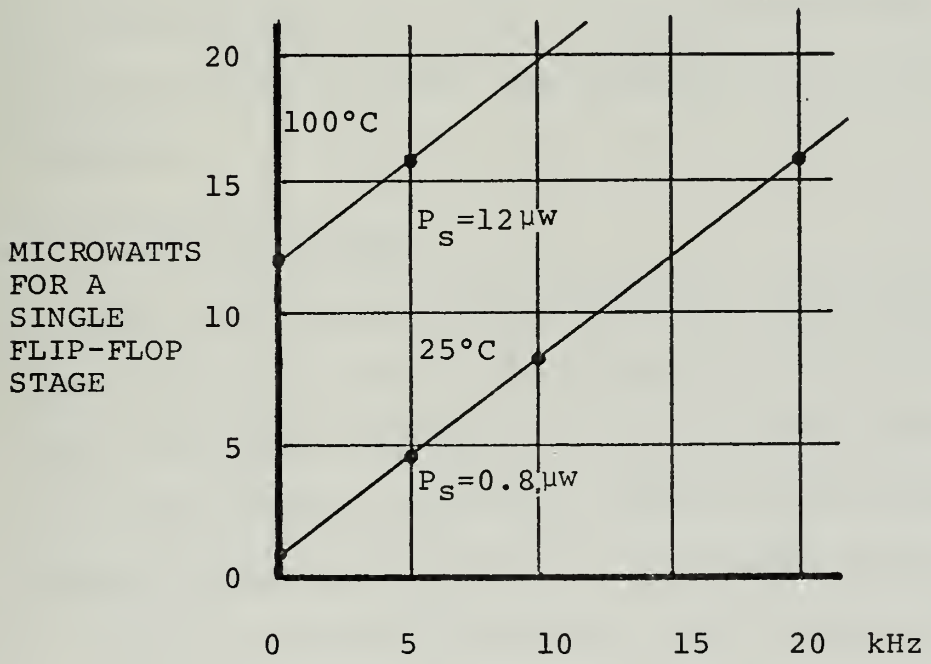


Figure 6. Power Dissipation as a Function of Repetition Rate

Since only a fixed number of operations are completed each minute in the detector, the clock rate will have no effect. The power economy is gained by reducing the total number of redundant operations. The repetition rate can go down to DC; however, the longest allowable high state of the clock input of the 64 bit shift registers can be no more than 100 microseconds. For this reason, the clock frequency was chosen to be 19.360 kHz which provides a good margin of safety.

III. HARDWARE IMPLEMENTATION

The logic design formulated in this section is organized according to the logic blocks diagrammed in Figure 1 of the Introduction. All logic functions are described and defined in the text and the figures. The later sections appraise the cost, size and power requirements of the resulting logic packages.

A. CLOCK AND TIMING

The first section of the detector, which sequences all operations and controls the functions of the device, is the clock and timing circuitry. Table I, which may be considered as the program, lists all sequences and defines the timing signals originating in the COS/MOS counter circuitry.

Since the COS/MOS inverters have nanowatt standby power and microwatt operational power, an excellent low power clock can be designed using the forward gain of an inverter with a suitable feedback arrangement. A good amplifier section for the oscillator is formed with an inverter stage and feedback resistor R_1 large enough (>10M) so that the attenuation and phase of the feedback are not appreciably affected. The resistor biases the output at about half the supply voltage.

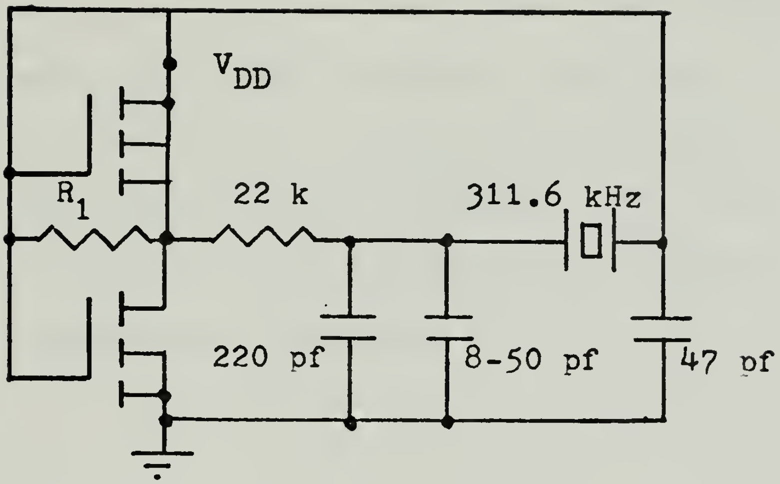


Figure 7. Crystal Oscillator Circuit Using RCA CD4007 Inverter

The crystal controlled phase shifting network is added in Figure 7 with provision for trimming to the design frequency. Using this circuit, stability should be on the order of 5 ppm and total supply current from a supply voltage of 4 volts is 50 microamps [Ref. 6].

Frequency division of the crystal oscillator is necessary due to duty cycle considerations which place the logic clocking frequency at 19,360 Hz as shown in Figure 8. One CD4004 provides four flip-flops in an asychronus counter to divide the 311.6 kHz crystal frequency down to the clocking frequency of 19,360 Hz. The bit counter consists of one-half of a CD4013 dual flip-flop plus a CD4022 eight bit divider/decoder to aid in decoding the timing signals. The word counter is another CD4022 connected to divide by five with a CD4001 gating package to reset the counter.

The sample counter is a CD4004 seven bit counter connected to count to 121, and ten other logic packages provide the necessary reset and decode functions. The second counter is a similar arrangement connected to reset at 60 using the same type logic packages. The first stage in the second counter is the recirculation counter; it divides by two to give a frequency of 1 Hz.

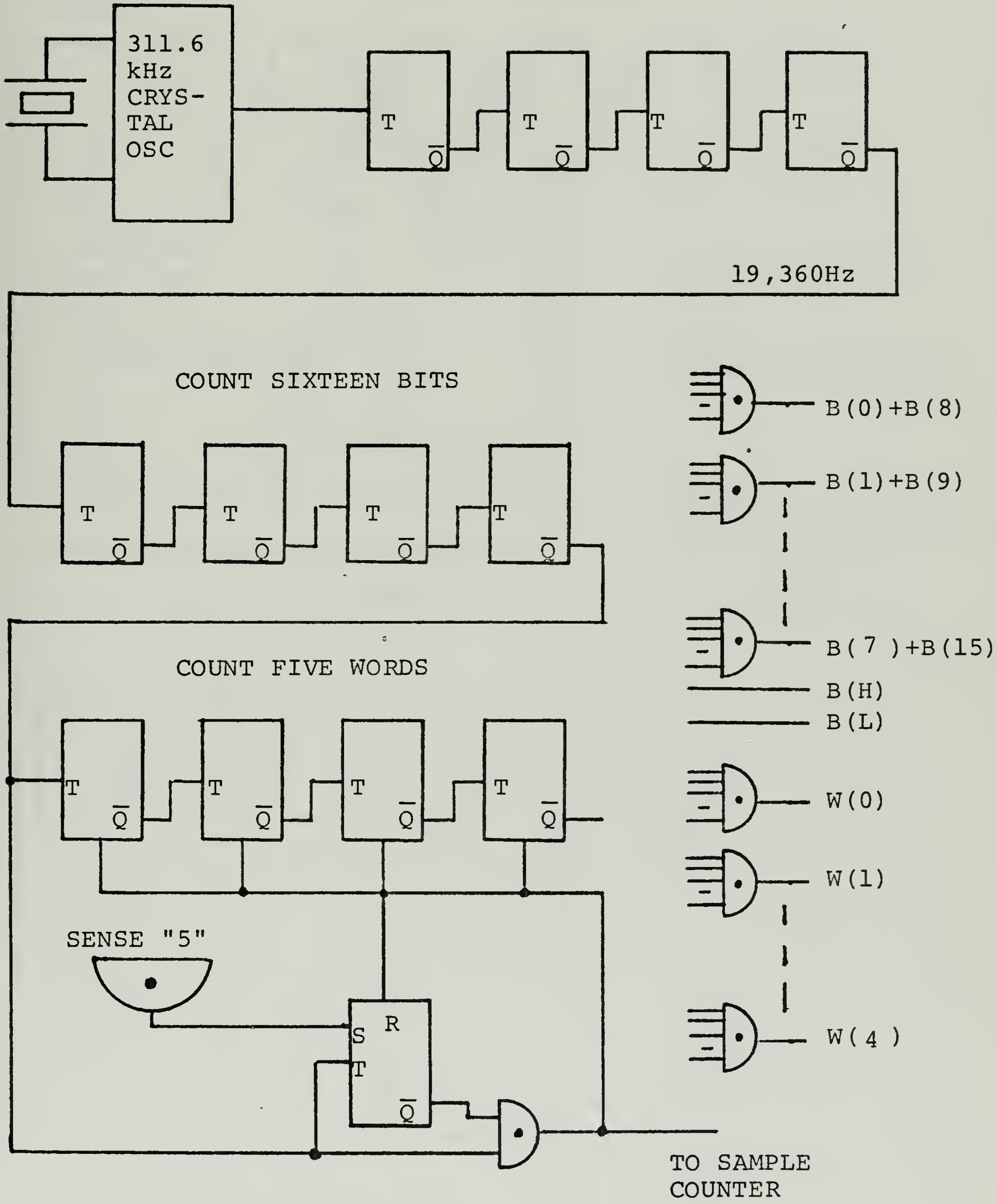


Figure 8. Clock - Bit and Word Counters Plus Decoders to Derive Timing Signals

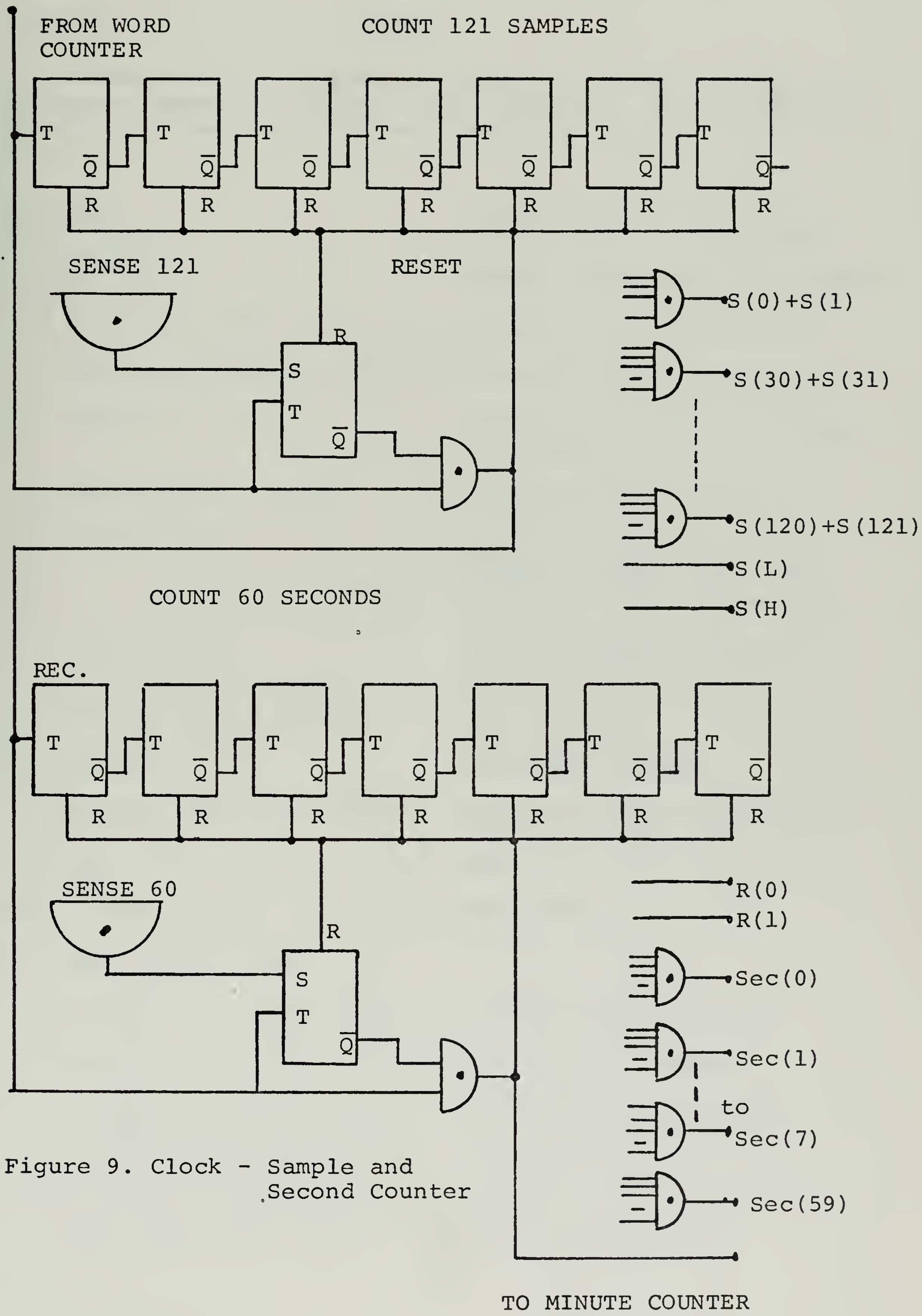


Figure 9. Clock - Sample and Second Counter

TABLE I. TIMING

These Boolean expressions define sequentially the timing signals referred to on the later diagrams. The gating circuitry for these expressions is shown in Figures 10,11,and 12.

$\mathrm{Sec}(0)\cdot R(0)$	RECIRCULATE BUFFER MEMORY
$\mathrm{Sec}(0)\cdot R(0)$	FIRST DIFFERENCE AND LIMITER
$S(L)\cdot E_D = \mathrm{Sec}(0)\cdot R(0)\left[S(0)+S(30)+S(60)+S(90)+S(120)\right]$	
$S(H)\cdot E_D = \mathrm{Sec}(0)\cdot R(0)\left[S(1)+S(31)+S(61)+S(91)+S(121)\right]$	
$E_D\cdot W(0)$	Clock 16 bit delay register and adder
$S(H)\cdot E_D\left[W(0) + W(2)\right]$	Shift Limiter register
$S(H)\cdot E_D\cdot W(1)$	Check difference and Set latch if good. Set sign change latch if necessary
$S(H)\cdot E_D\cdot W(2)$	Shift present difference to temporary storage.
$S(H)\cdot E_D\cdot W(2)\cdot B(15)$	Change sign.
$\mathrm{Sec}(1)\cdot R(0)$	TEMPORARY STORAGE
$E_D\cdot W(2)\ \left[B(8) \text{ to } B(15)\right]^*$	Shift in 8 bit difference
$E_{sum}\cdot W(0)$	Shift out difference
$\mathrm{Sec}\ (2)\cdot R(0)$	TIDE FILTER
$E_{sum} = \mathrm{Sec}(2)\cdot R(0)$	
$E_{sum}\left[W(0)\ B(5) \text{ to} B(15)\right]^* + W(1)$	Multiply by -3.920392
$E_{sum}\left[W(0)\ B(7) \text{ to} B(15)\right]^* + W(1)$	Multiply by 5.841800
$E_{sum}\cdot\left[S(1) + S(121)\right]\cdot\left[W(1)\cdot B(4) \text{ to } B(12)\right]^*$	Add differences 1 and 121
$E_{sum}\cdot\left[S(1)+S(121)\right]\cdot W(1)\cdot B(12)$	Set sign latch
$E_{sum} + \left[E_{sq}\cdot W(1)\cdot B(0)\right]$	Shift accumulator

TABLE I. Continued

Sec(3) R(0)	SQUARE
$E_{sq} = \text{Sec}(3) \cdot R(0) \cdot [S(0) \text{ to } S(75)]^{*}$	
$E_{sq} \cdot W(1) \cdot B(0)$	Shift accumulator in sum and check bit for "one"
$E_{sq} \cdot W(0)$	Shift input store and SQUARE accumulator
$\text{Sec}(4) \cdot R(0)$	INTEGRATE
$E_{int} = \text{Sec}(4) \cdot R(0) \cdot S(0)$	
$E_{int} \cdot W(0) [B(11) \text{ to } B(15)]$	Shift in the squared value
$E_{int} \cdot W(1)$	Shift registers
$\text{Sec}(59) \cdot R(0)$	INPUT SECTION
$E_{in} = \text{Sec}(59) \cdot R(0) \cdot S(121)$	
$E_{in} \cdot W(0)$	Shift in pressure value
$E_{in} \cdot W(1)$	Shift in temperature, reset pressure counter
$E_{in} \cdot W(2)$	Shift in current_u, reset temperature
$E_{in} \cdot W(3)$	Shift in current_v, reset current_u
$E_{in} \cdot W(4)$	Shift in time, reset current_v

* $B(i) \text{ to } B(j) = B(i) + B(i+1) + \ldots + B(j)$

The decoding circuitry for these signals are shown together in Figures 10,11, and 12. Usually they will be located on the same board as the device they control.

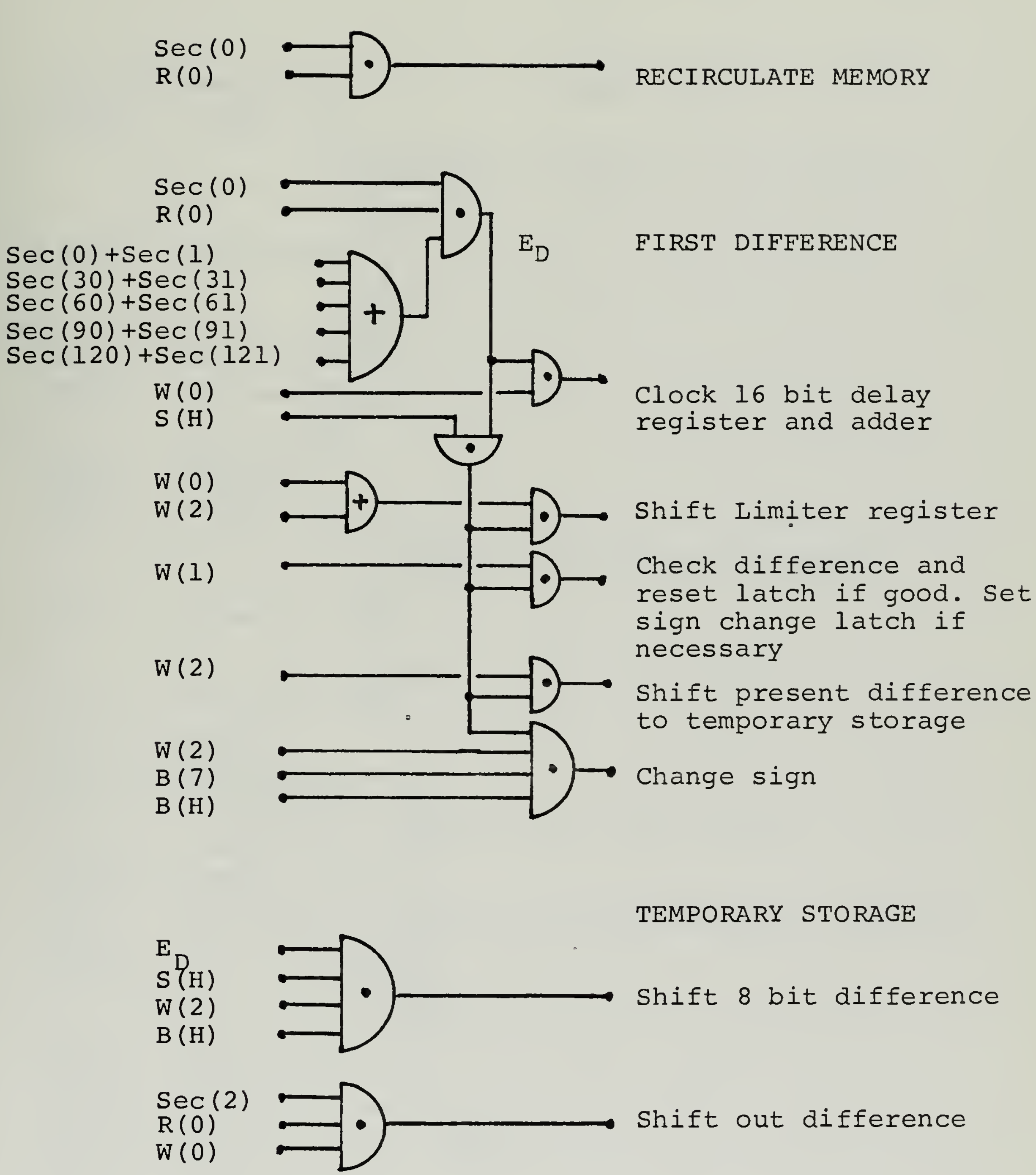


Figure 10. Timing Decoding Gates

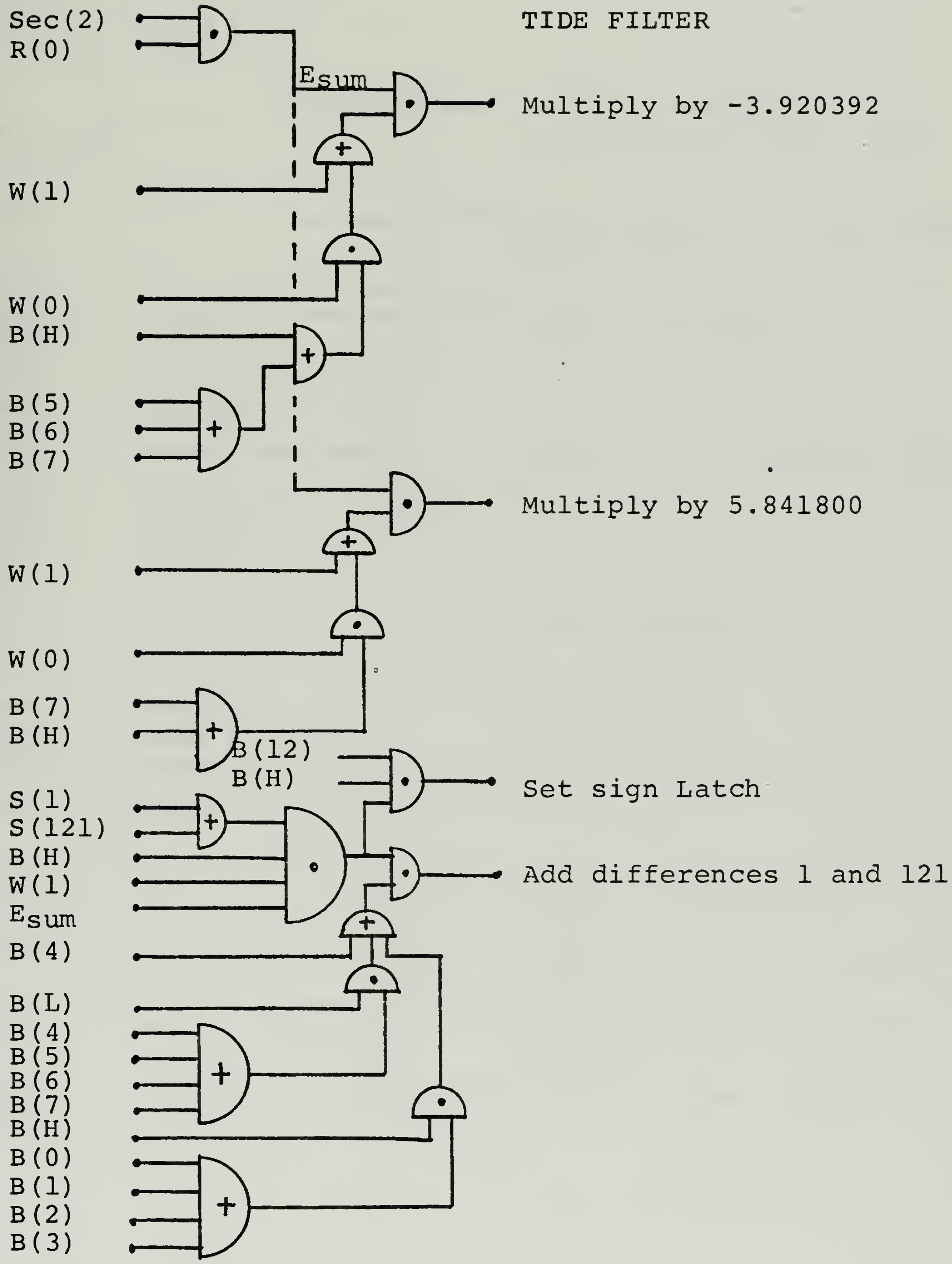


Figure 11. Timing Decoding Gates

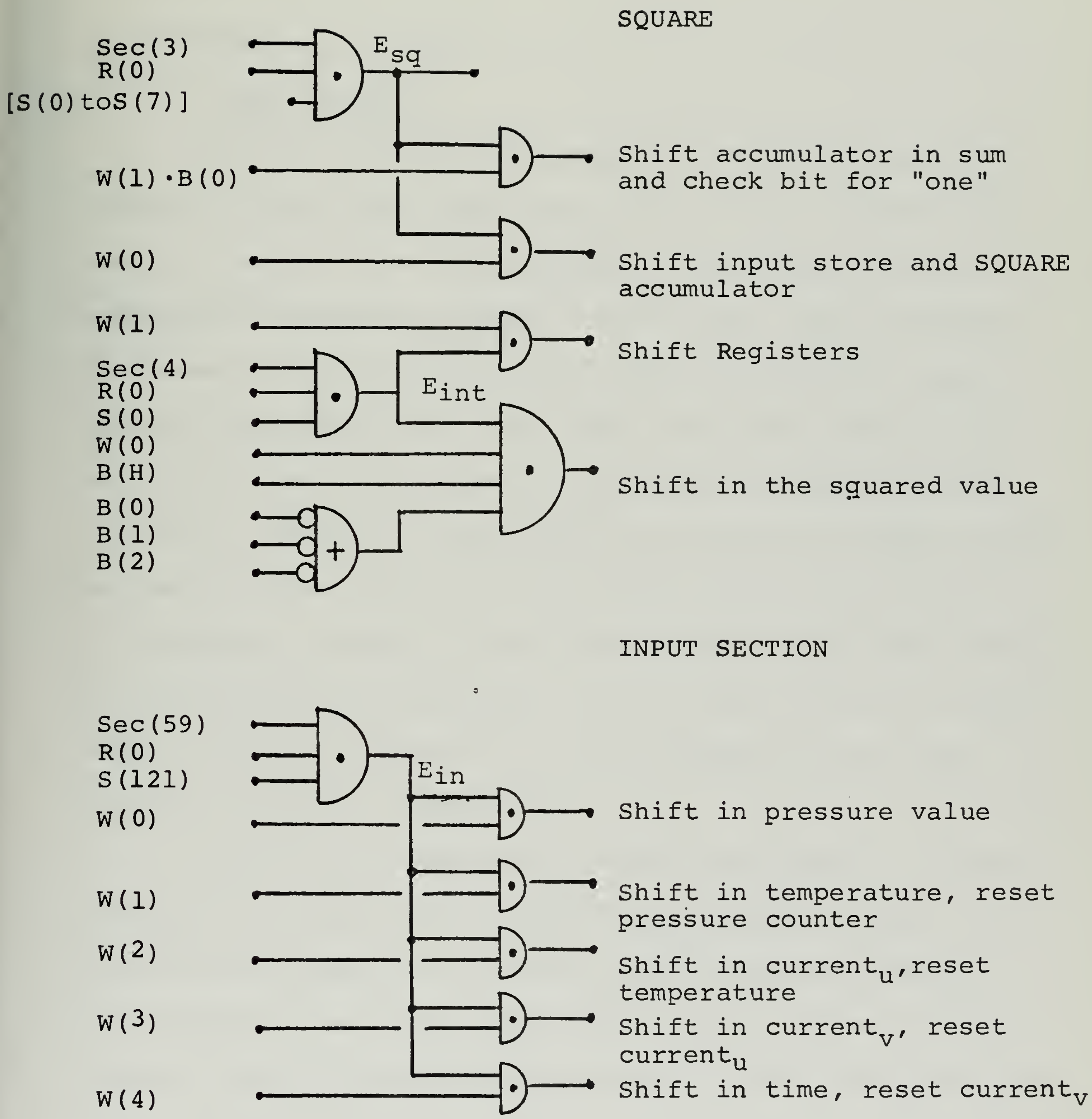


Figure 12. Timing Decoding Gates

B. PROCESSING CIRCUITS

1. Input Section

The five inputs consisting of variable frequency signals are used to either count or gate five sixteen bit binary counters. The fifth counter is provided with the additional circuitry so that it will count time in minutes. On command from the system clock each of the five counters sequentially shifts its value into the system memory and, except for the time channel, resets for the next counting interval. Figure 13 shows only the pressure and time channel because all five inputs are the same except for the gating on the minute counter. Since the two-hour sequencing signal for the detector is obtained from the time counter, it must reset at 65,520, a multiple of 120 minutes, so that the two-hour sequence is maintained.

All of the counters are made from CD4004's. Three counters for each input provide the necessary sixteen bits. Two eight bit parallel to serial shift registers convert the counter value into serial form for the system's buffer memory. Fourteen additional packages provide the gating and mode control circuitry.

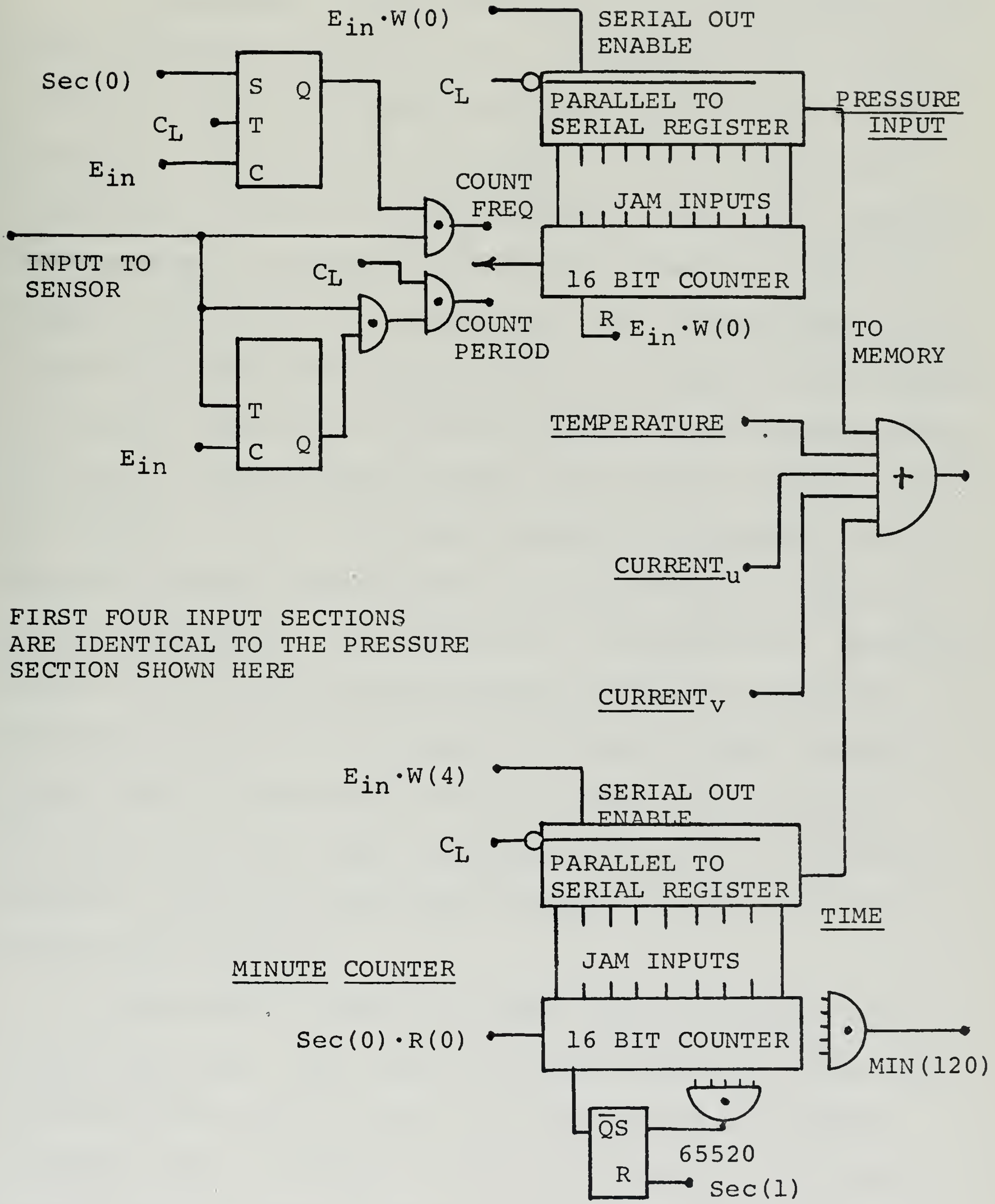
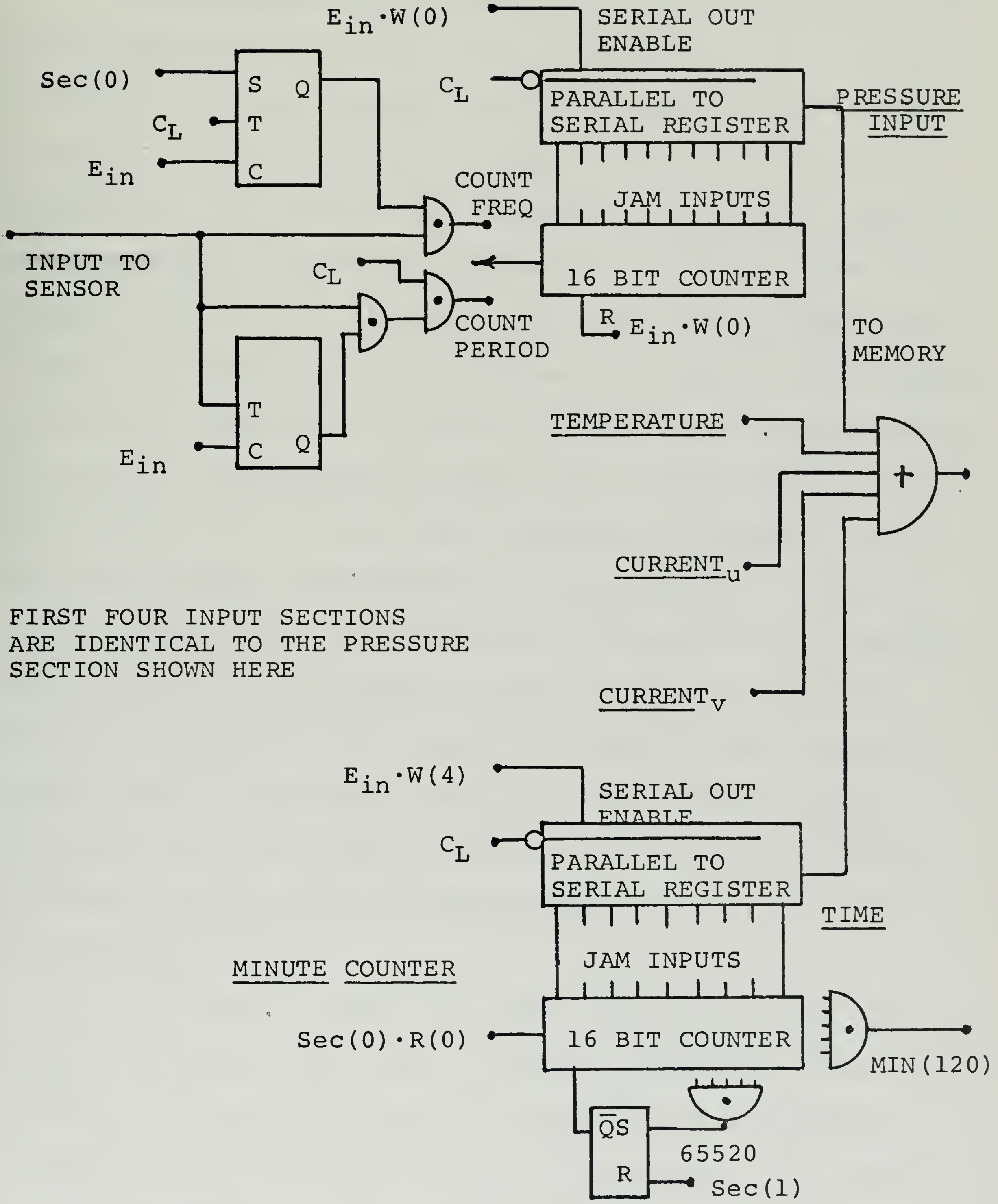


Figure 13. Input Circuitry - Each Output Can be Wired to Either Gate or Count the 16 Bit Counters. Thus, the Mean Frequency or Mean Period of the Input Is Shifted to the 2 Hour Buffer Memory Once Each Minute



FIRST FOUR INPUT SECTIONS ARE IDENTICAL TO THE PRESSURE SECTION SHOWN HERE

Figure 13. Input Circuitry - Each Output Can be Wired to Either Gate or Count the 16 Bit Counters. Thus, the Mean Frequency or Mean Period of the Input Is Shifted to the 2 Hour Buffer Memory Once Each Minute

2. System Memory

The system memory consists of a discrete state static shift register of 9680 bits in a recirculation loop storing the past 121 minute samples organized sequentially by words and bits. The bit, word and sample counters provide for addressing the data; further division of the clock frequency into seconds and minutes provides all necessary timing signals for the rest of the detector. The timing signals are obtained by coincidence gating across the stages of the asychronus binary counters. In order to reduce the transient power, the data is recirculated through the register length only once per minute. On this one pass, clocking and gating read out the ten pressure words needed for computation of the five first differences. The 9680 shifts averaged over a minute's time results in an average clocking frequency of 150 Hz. From data similar to the graph of Figure 6, the transient power used in the memory should be about 100 microwatts; however, the 64 bit shift registers are still in the developmental stage and this power estimate is meant only as a comparative estimate.

In order to obtain the 9680 bit length in COS/MOS logic, 152 logic chips must be series connected. The exact 9680 bits required is made of 151 sixty-four bit TA5989 registers and one CD4006 sixteen bit register. The total memory package should dissipate about 0.3 mW and fill a volume of 0.34 cu.ft. on six standard 2.5 by 5.0 inch logic cards.

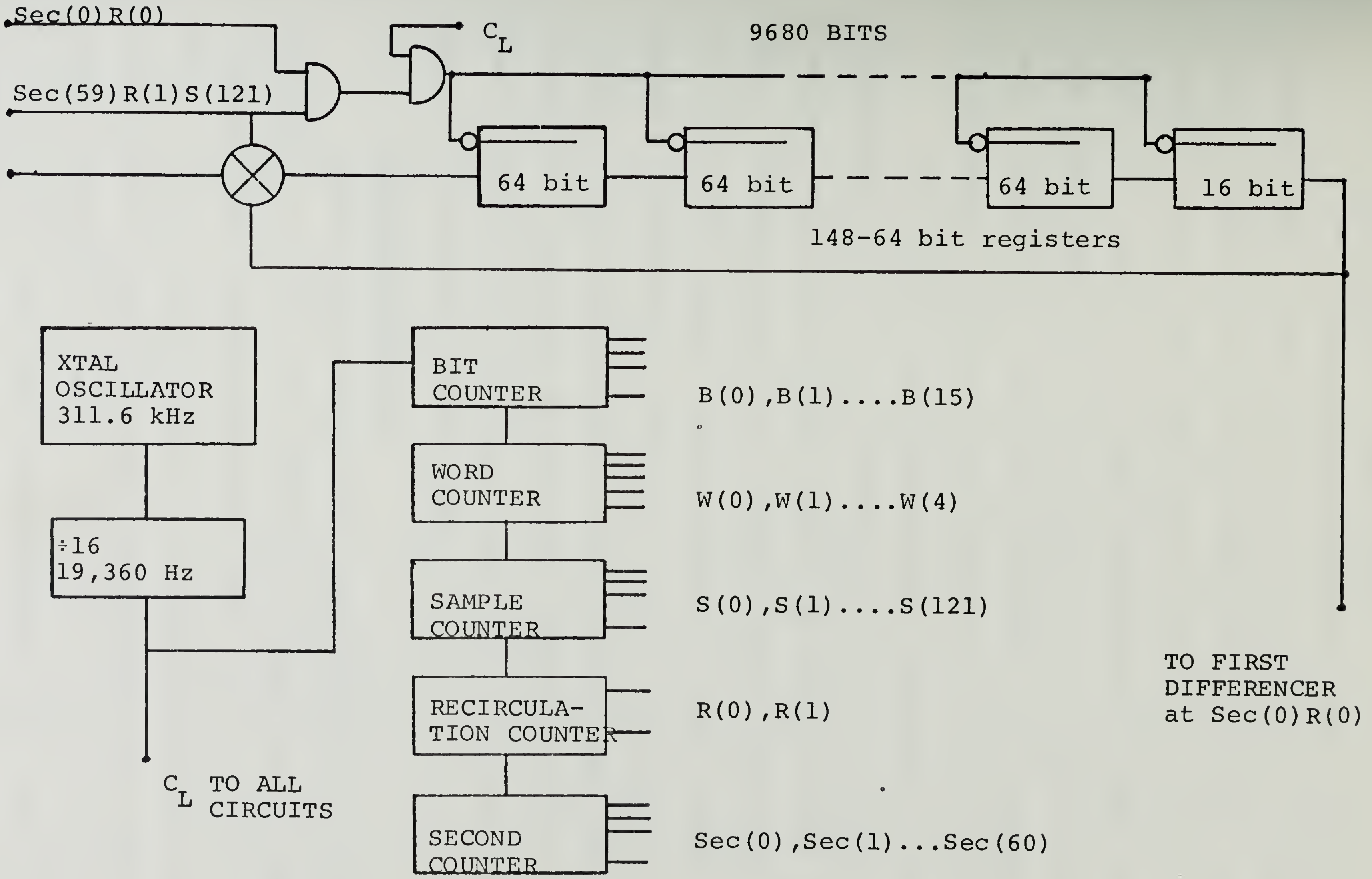


Figure 14. Two Hour Temporary Storage Memory.Contains Past 2 Hours of Sample Values in a Static Shift Register. Total Power is Less Than 0.3 MW.

3. First Difference and Limiter

First differences are calculated once a minute on the five sample pairs S(0), S(1); S(30), S(31);...S(120), S(121) by delaying the first sample in a register, complementing the next and adding the two serially. Since the original input number is allowed to overflow the input counters, the signed result must be corrected if the sign is inconsistent. Gating also checks the difference for unreasonable values before allowing them to pass on to temporary storage.

The sixteen bit delay and all the eight bit storage requirements in the first differencer can be obtained from various connections of the CD4006, an eighteen stage register. Two CD4015A dual four stage parallel output registers provide the necessary connections used in the limiter to sense and correct the difference value. One-third of a TA5963 triple serial adder is used for the adder and a CD4013 dual flip-flop performs the complement function. Twelve additional packages as listed in Table II provide all the gating and control logic.

4. Temporary Storage

Since only correct first differences are allowed through the limiter, the temporary storage must be capable of nondestructive read-out when the tide filter calls for inputs. This is a very simple matter of circularly shifting the data to the right. Only the 8 most insignificant bits need to be stored, because the average first difference is on the order of 4 bits. The validity of this assumption was verified by simulation.

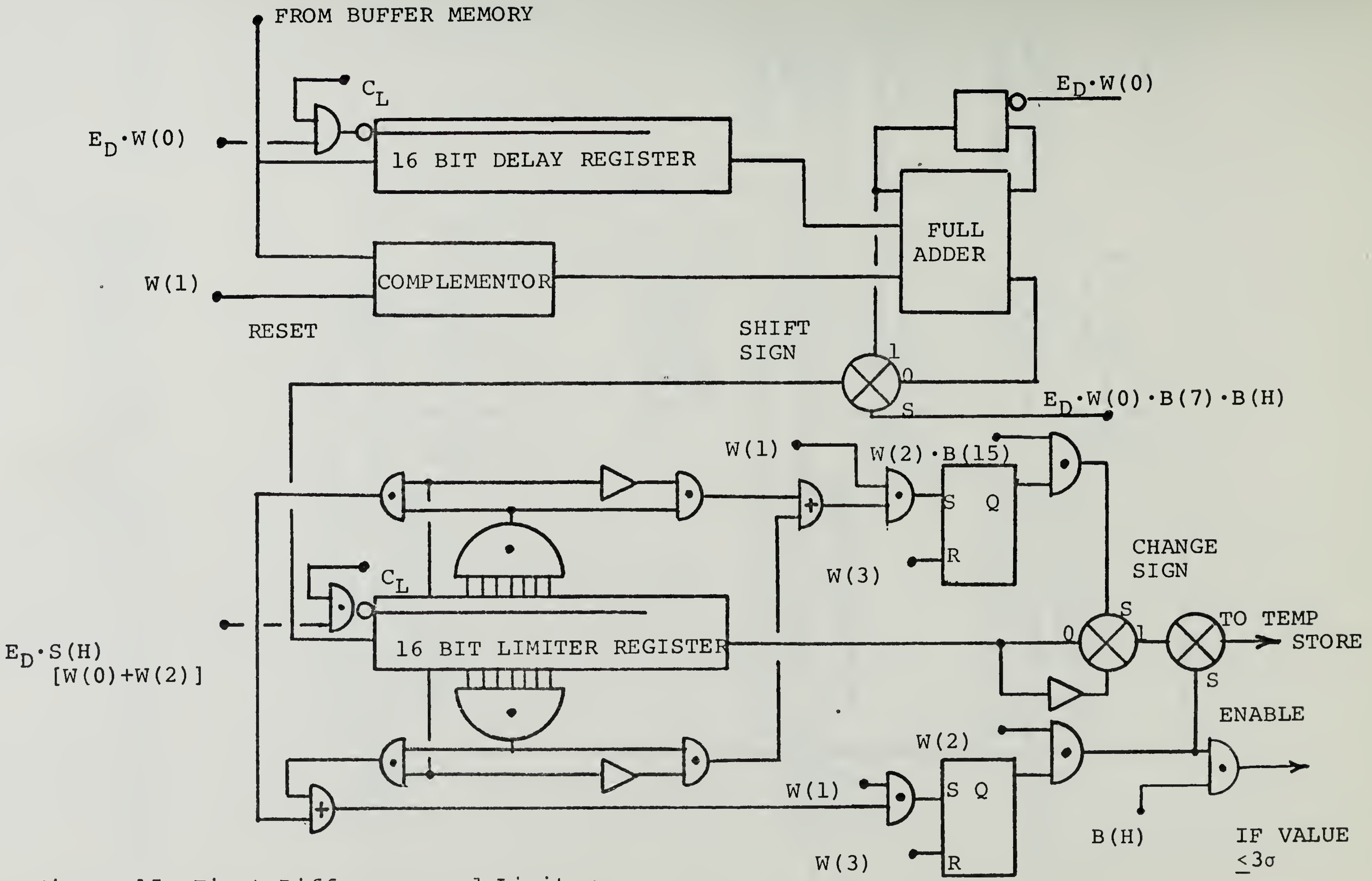


Figure 15. First Difference and Limiter

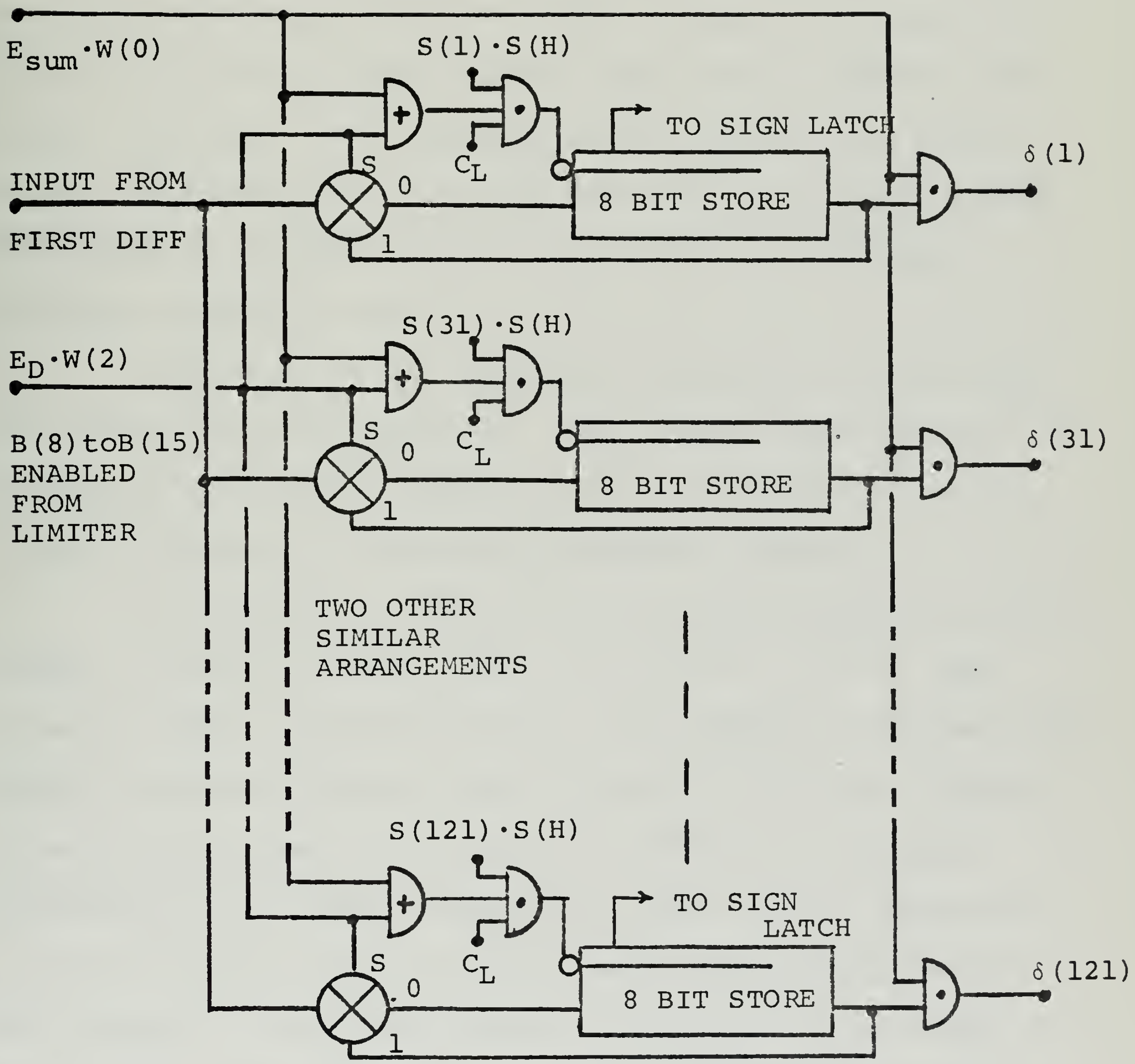


Figure 16. Temporary Storage - Five 8 Bit Registers Store the Previous Differences in Case the Present One Is Rejected by the Limiter

5. Tide Filter

The five differences temporarily stored are simply multiplied by appropriate weighting values and summed to obtain the filtered output. Since there are only three different constants, the differences are gated to the proper section through serial switches and the sum of these values is stored in a register which also serves as the input register to the squarer.

Since one of the weighting constants is unity, only two multipliers are required. They simply shift and add the input the appropriate number of times and proper clocking keeps the result in the same sixteen bit format.

The constant -3.920392 can be approximated with the number -3.920410 or $2^2 + (2^{-4} + 2^{-6} + 2^{-10} + 2^{-11})$. This constant, when multiplied by the input number of sixteen bits gives a possible maximum length result of 29 bits. Through simulation and device consideration, however, the input difference will rarely exceed eight bits and the approximation to reality of the constants makes the significance of the lower five bits questionable. Truncation to sixteen bits before addition, then, will have no effect. Similarly, 5.84180 is approximated by 5.842797 or $2^3 - (2^1 + 2^{-3} + 2^{-5} + 2^{-9})$.

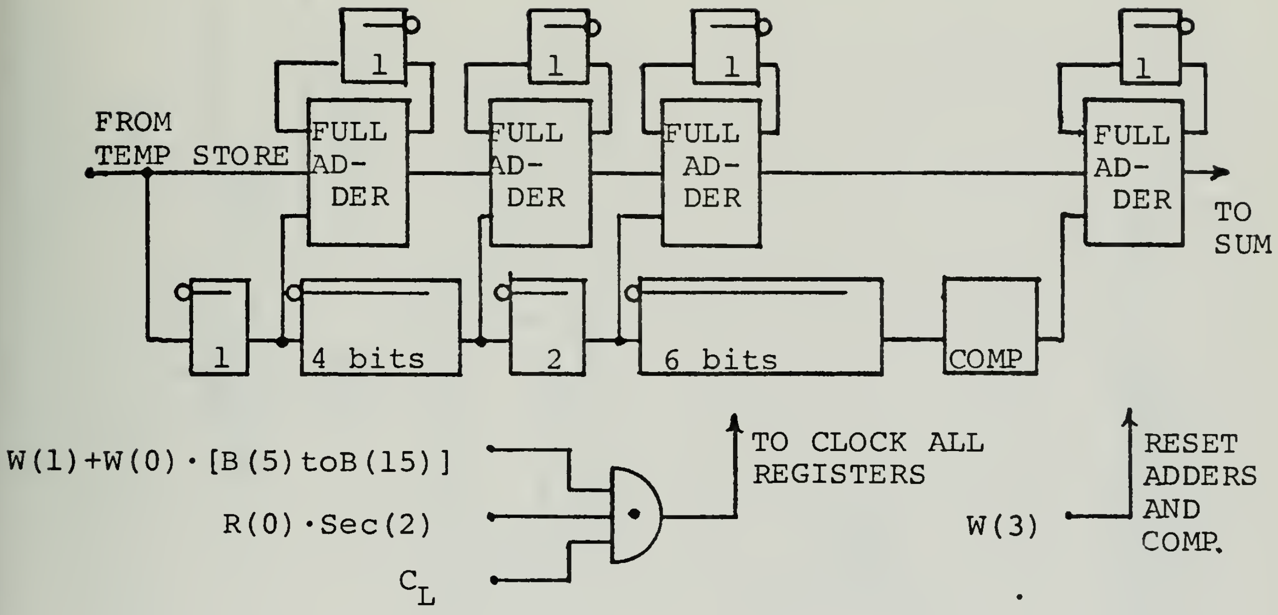
The maximum of the sum of the five weighted differences will be approximately the sum of the magnitudes (1 + 4 + 6 + 4 + 1) times the average input value. Limiting the maximum first difference to seven bits or 128_{10} requires

eleven whole number bits in the accumulator leaving four fractional bits plus the sign bit. To multiply by the constant one, the input need only be left-shifted four bits to account for the change in the binary point. Clocking and gating accomplish the necessary four bit delay.

The analytic two's complement code used throughout the detector requires no special handling in the multipliers; however, since the eight bit differences are to be extended to sixteen bit precision words in the accumulator, care must be taken to pack the additional eight created bits with zeros or ones depending on the sign of the original number. The sign latch in Figure 18 tests the sign from the 1st and 121st difference and ensures that the input is held at the proper level if there are no incoming bits.

All of the timing signals referenced on the figures are defined in Table I. The Figures 10, 11 and 12 diagram how the timing signals are derived through gating and decoding the frequency divider outputs.

The total number of twenty-five delays in the two multipliers is provided by four CD4015 dual four bit registers. Three more TA5963 triple serial adders perform the summing functions, with the result stored in an additional two CD4015's. The two complementors are similar to the one used in the first difference and diagrammed in Appendix A.



MULTIPLIER X -3.920392 APPROXIMATED IN BINARY TO 26 BIT PRECISION BEFORE TRUNCATION TO 16

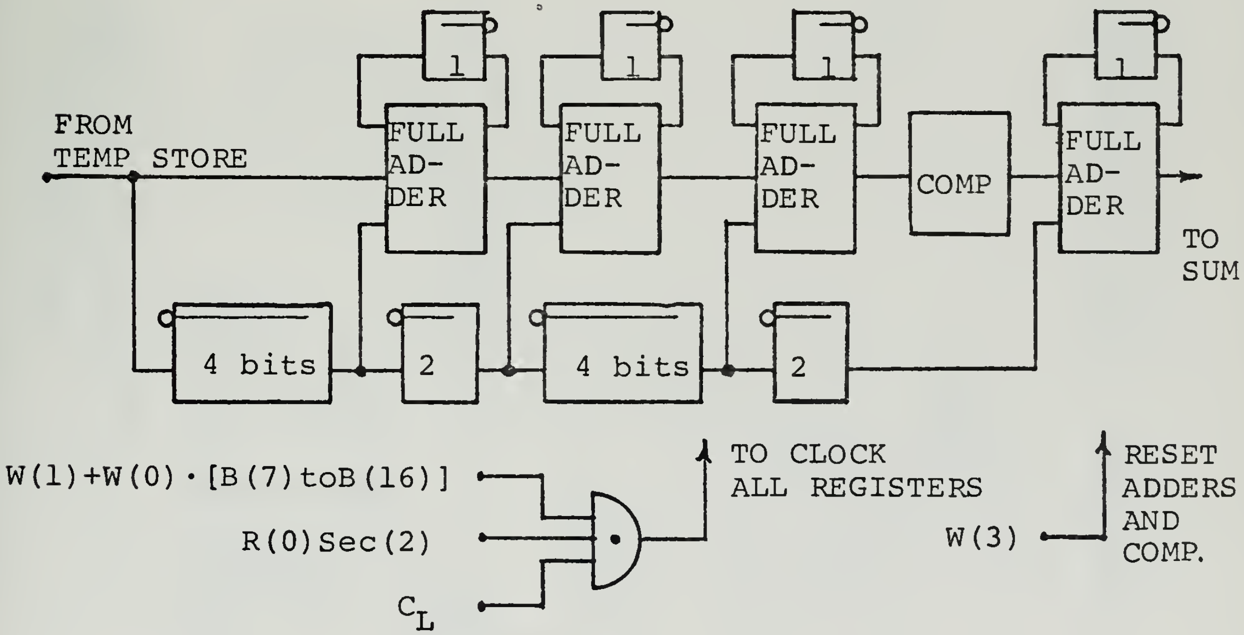


Figure 17. Multiplier X 5.841800 APPROXIMATED IN BINARY TO 26 BIT PRECISION BEFORE TRUNCATION TO 16

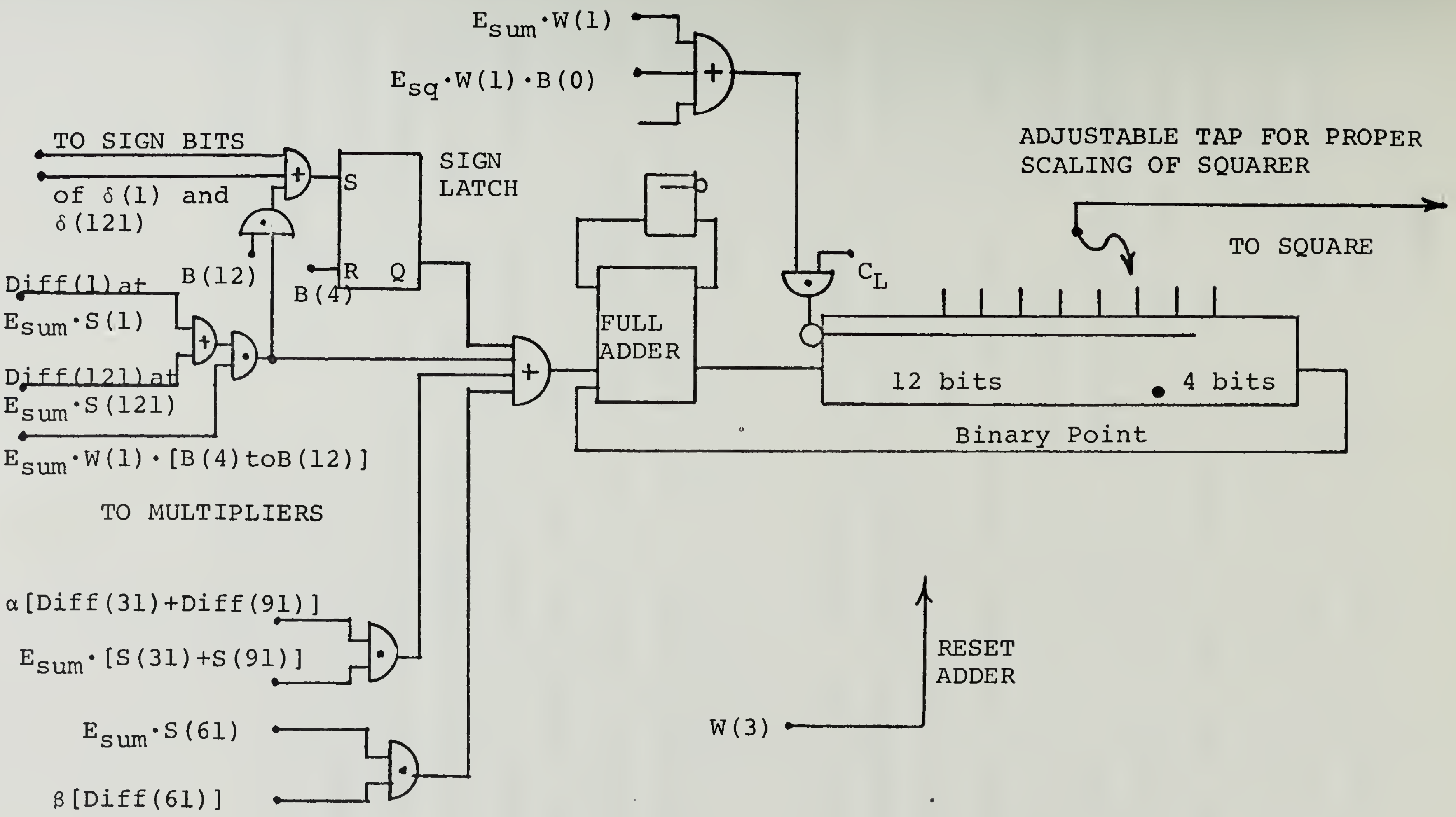
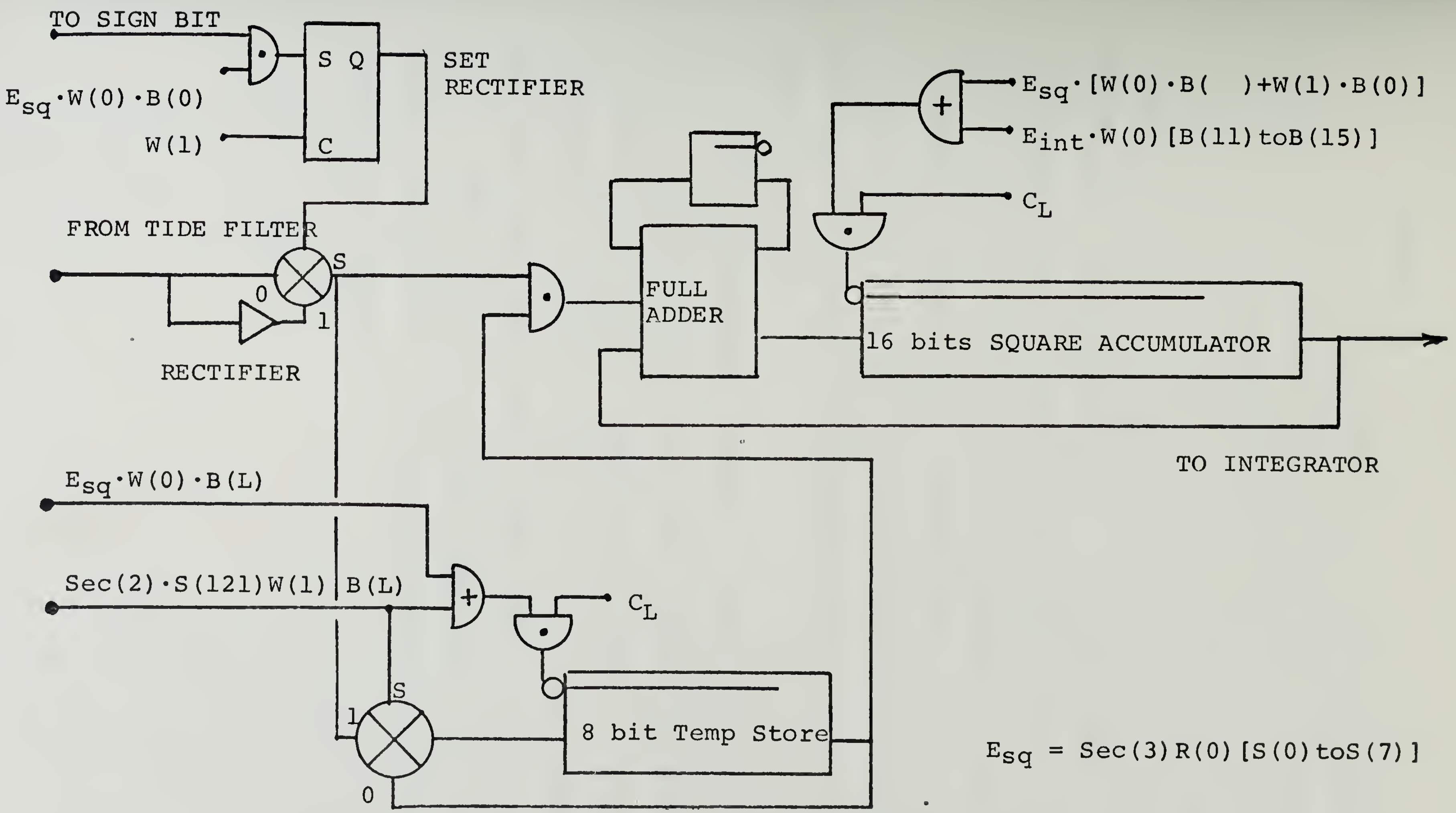


Figure 18. Accumulator. Each of the Weighted Differences Are Added into the Accumulator maintaining 4 Fractional Bits

6. Squaring

The squaring package observes the value stored in the sum accumulator of the filter and shifts and adds it into the output storage. The square circuitry would handle negative as well as positive numbers if care were taken to provide all the necessary ones in the extra negative bits. A simpler method to implement, however, is to simply rectify the input. By testing the sign bit in the sum accumulator before each operation, negative numbers can be complemented bit by bit. This method, as shown in Figure 19.,gives an error of one in the least significant bit because it takes the one's instead of two's complement. The error is insignificant.

For each bit the input is shifted and if the current bit is a 1, the shifted value is added to the result. The sum in the output of the filter contains a sign and four fractional bits; in order to make the detector adaptable to a variety of dynamic ranges, the tap on this register is made adjustable. The input to the squarer can only be eight bits in length, but the adjustable tap gives a range of the most sensitive eight bits from 1/32 to 15 to the maximum values between 16 and 65,536. Simulation later showed the most probable tap to be at the binary point.



$E_{sq} = Sec(3) R(0)\,[S(0) to S(7)]$

Figure 19. SQUARER

Squares 8 Bit Value from Tide Filter by Shifting One Bit, Testing for a "1" and Adding Incrementally to the Accumulator

7. Integrator

Since the squared output is to be integrated over a decaying time constant of sixty minutes, the maximum available storage must be about thirty times the maximum value to ensure against overflow. To stay within the sixteen bit precision, the input squared value must be truncated to the eleven most significant bits. The decay factor of .991666 is accomplished by delaying seven bits and then subtracting. Taps are accessible on the delay register to vary the time constant of integration. Coincidence gating on the storage register synchronized to the clock determines the value in the tsunami detector and provides the control logic with its go/no-go signal.

The necessary storage for the squarer and integrator can all be formed from the CD4006 eighteen stage register. A CD4015 dual four bit register provides for the multiplication in the integrator along with two of the summers left over from the triple serial adder used in the first differencer. The complementor again comprises a dual flip-flop CD4013 and nine additional gating packages complete the squarer and integrator.

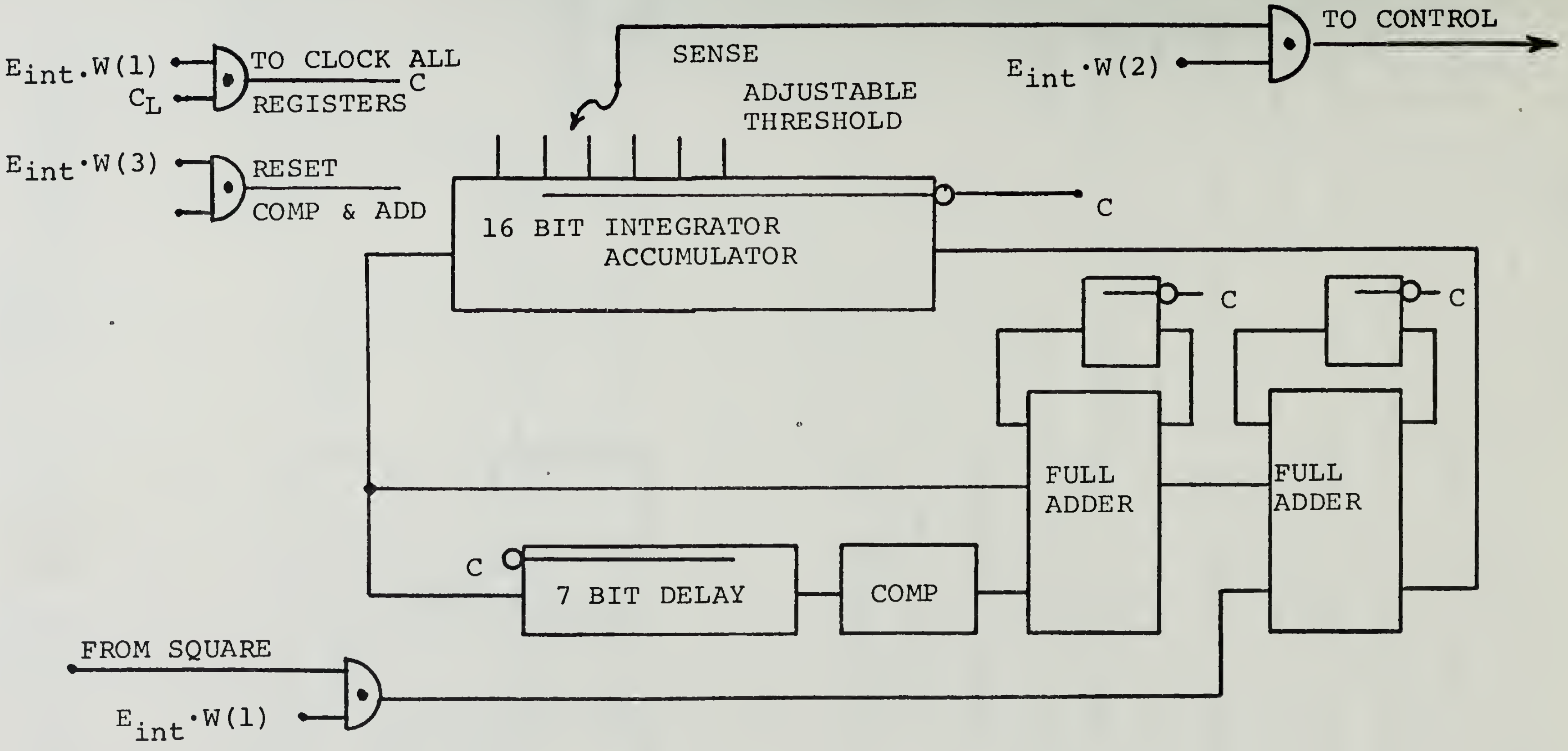


Figure 20. Integrator - 7 Bit Delay Gives .99166 Decay Factor But Other Delays Are Accessible with Solder Jumpers. Threshold of Detection Also Adjustable

8. Storage Control

This section of the detector is designed with flexibility as the prime consideration since the actual device used for long term storage will be an independent system. Whether a tape system or core storage is used the only output required of the detector is a yes or no answer every two hours qualified by considerations of its previous answers. A critical,or event, is defined whenever the value in the Integrator register exceeds the threshold set by the sense tap. A store command is given under the following conditions:

a. A critical is detected and the past n consecutive hours have not been critical;

b. A critical is not detected and the past two hours were critical.

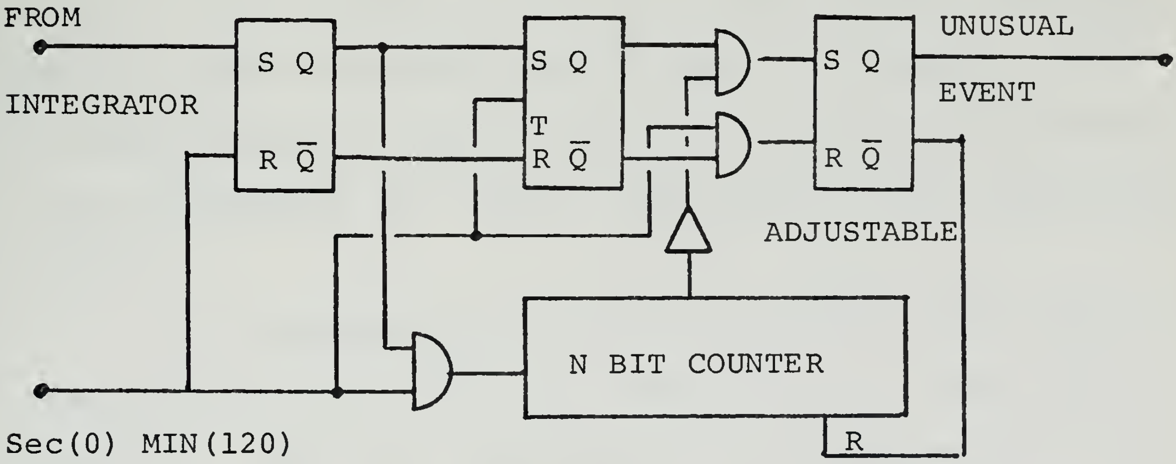


Figure 21. Control - Three Flip-Flops and a Counter Provide Logic to Determine if an Unusual Event Should Be Signaled

C. PACKAGE CONSIDERATIONS

The COS/MOS line offers sufficient complexity per chip to keep the overall package count low, and also, enough variety to provide for optimum utilization of the available circuitry. The total package requirements for each of the sections of the detector are listed in Table II, with a cost estimation in Table III.

1. Space and Cost

The size and cost predictions of the detector are based on an experimental set-up which will use dual in-line packaging on a general purpose bread board arrangement. Standard Logic Inc. offers a line of logic cards which have various combinations of fourteen and sixteen pin DIP sockets and all mounting and interconnecting hardware. Computer aided wiring and interconnections complete the system. Using the Standard Logic boards from 20 to 30 DIP packages depending on the number of sixteen and fourteen pin packages, can be arranged on each 2.5 by 5.0 inch card. Figure 22 shows the logic arrangement and overall dimensions of this experimental set-up.

For reliability the detector should be made with flat-packs on printed circuit boards with all solder connections. The space requirements will be approximately 40 to 60 percent of the experimental arrangement due to the elimination of the DIP sockets and the interconnecting wiring.

TABLE II. PACKAGE REQUIREMENTS

FUNCTION	FIGURE	NUMBER	LOGIC CIRCUIT
CLOCK	8	2	CD4022 Divide by 8 Counter/Decoder
		1	CD4004 Seven Stage Counter
		1	CD4013 Dual D Flip-flop
		1	CD4009 Hex Inverter
	9	11	CD4023 Triple 3-Input NAND Gate
		4	CD4001 Quad 2-Input NOR Gate
		2	CD4009 Hex Inverters
		2	CD4004 Seven Stage Ripple Counters
		1	CD4013 Dual D Flip-flop
		25	total
INPUT	13	15	CD4004 Seven Stage Ripple Counter
		10	CD4014 8-stage Parallel in/Serial out
		5	CD4011 Quad 2-Input NAND Gate
		4	TA5872 Dual J K Flip-flop
		2	CD4023 Triple 3-Input NAND Gate
		1	CD4000 Dual 3-Input NOR plus Inverter
		1	CD4013 Dual D Flip-flop
		1	CD4009 Hex Inverter
		39	total
MEMORY	14	151	TA5989 64-Stage Static Shift Register
		2	CD4011 Quad 2-Input NAND Gate
		1	CD4006 18 stage Static Shift Register
		154	total

TABLE II. Continued

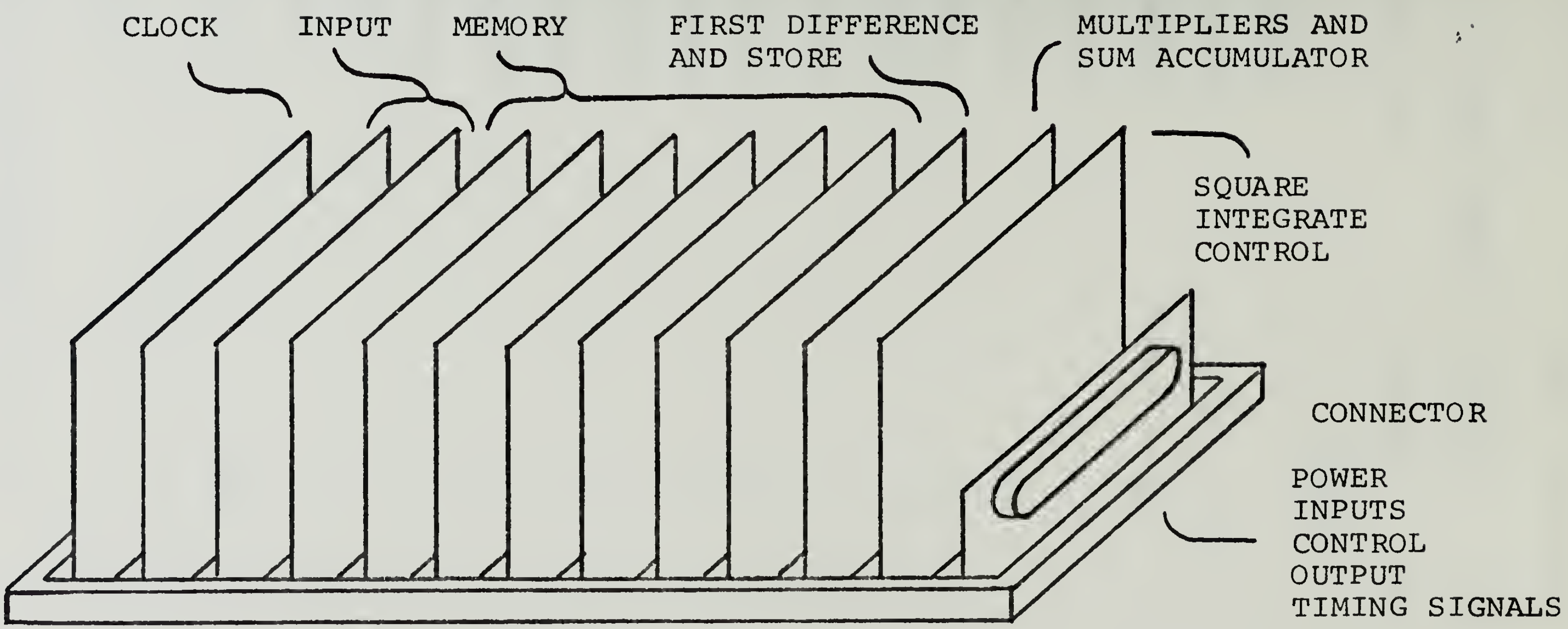
FUNCTION	FIGURE	NUMBER	LOGIC CIRCUIT
FIRST DIFFERENCE	15	5	CD4011 Quad 2-Input NAND Gate
		2	CD4013 Dual D Flip-flop
		2	CD4015 Dual 4-Stage Parallel out
		2	CD4016 Quad Bilateral Switch
		2	CD4012 Dual 4-Input NAND Gate
		2	CD4009 Hex Inverter
		1	CD4006 18 Stage Static Shift Register
		1	TA5963 Triple Serial Adder
		17	total
TEMPORARY STORE	16	3	CD4006 18-Stage Static Register
		3	CD4011 Quad 2-Input NAND Gate
		3	CD4016 Quad Bilateral Switch
		2	CD4009 Hex Inverter
		2	CD4023 Triple 3-Input NAND Gate
		13	total
MULTIPLIERS	17	4	CD4015 Dual 4-Stage Parallel out
		3	TA5963 Triple Serial Adders
		2	CD4011 Quad 2-Input NAND Gate
		2	CD4013 Dual D Flip-flop
		1	CD4023 Triple 3-Input NAND Gate
		13	total

TABLE II. Continued

FUNCTION	FIGURE NUMBER	LOGIC CIRCUIT
SUM ACCUMULATOR	18	2 CD4011 Quad 2-Input NAND Gate
		2 CD4015 Dual 4-Stage Parallel out
		1 TA5963 Triple Serial Adder
		1 CD4023 Triple 3-Input NAND Gate
		1 CD4000 Dual 3-Input NOR plus Inverter
		1 CD4002 Dual 4-Input NOR Gate
		8 total
SQUARE	19	2 CD4006 18-Stage Static Register
		2 CD4011 Quad 2-Input NAND Gate
		1 CD4001 Quad 2-Input NOR Gate
		1 CD4009 Hex Inverter
		1 CD4016 Quad Bilateral Switch
		1 TA5963 Triple Serial Adder
		8 total
INTEGRATE	20	3 CD4015 Dual 4-Stage Parallel out
		2 CD4011 Quad 2-Input NAND Gate
		1 CD4013 Dual D Flip-flop
		1 TA5963 Triple Serial Adder
		7 total

TABLE II. Continued

FUNCTION	FIGURE	NUMBER	LOGIC CIRCUIT
STORAGE CONTROL	21	2	CD4011 Quad 2-Input NAND Gate
		1	CD4004 Seven Stage Ripple Counter
		1	CD4013 Dual Flip-flop
		1	CD4009 Hex Inverter
		5	total
TIMING GATES	10 11 & 12	6	CD4011 Quad 2-Input NAND Gate
		2	CD4001 Quad 2-Input NOR Gate
		2	CD4023 Triple 3-Input NAND Gate
		2	CD4025 Triple 3-Input NOR Gate
		1	CD4002 Dual 4-Input NOR Gate
		1	CD4012 Dual 4-Input NAND Gate
		2	CD4009 Hex Inverter
		16	total



CLOCK	30 DIPS	1 CARD
INPUT	45 DIPS	2 CARDS
MEMORY	153 DIPS	6 CARDS
1ST DIFFERENCE AND TEMPORARY STORAGE	30 DIPS	1 CARD
MULTIPLIERS AND SUM ACCUMULATOR	26 DIPS	1 CARD
SQUARE INTEGRATE AND CONTROL	25 DIPS	1 CARD

Figure 22. Physical Configuration

TABLE III. COST EVALUATION

Low Voltage Series CD4000A 3 to 15 volts

Flat Pack -55°C to +125°C

	Number	Cost Each	Total Cost
CD4000AK	2	$ 7.00	$ 14.00
CD4001AK	6	7.40	44.40
CD4004AK	19	14.75	280.25
CD4006AK	7	18.00	126.00
CD4009AK	12	11.00	132.00
CD4011AK	31	7.40	222.00
CD4012AK	3	7.65	22.95
CD4013AK	9	10.05	90.45
CD4014AK	10	20.15	201.50
CD4015AK	11	20.25	222.75
CD4016AK	6	10.80	64.80
CD4022AK	2	20.95	41.80
CD4023AK	19	7.65	145.35
CD4025AK	2	7.65	15.30
TA5872		10.00	30.00
TA5963	7	14.00	100.00
TA5989	151	20.00	3020.00
TOTAL			$4752.55
Similar arrangement in Dual In Line Plastic Packages		Approx-	$2000.00

-40° to +75°C

2. Power

The actual quiescent power required varies considerably from chip to chip and is very temperature dependent. In most cases, RCA specifications include a "typical" and "maximum" value. The estimates given assume the "typical" measurement and an ambient temperature of 25°C. The actual power at the ocean floor, therefore, should be less than the estimate. Also the transient power is of no significance in the digital processing circuitry because the maximum number of operations of any of the circuits is still on the order of 160 per minute. This gives an average repetition rate of only 3 Hz when averaged over the one minute period. For this reason only the transient power of the clock, the decoder and the buffer memory are included.

The total power drain of 0.5 mW required for the whole detector system is almost three orders of magnitude smaller than a comparable logic package of conventional circuitry. Compared to the power requirements of the sensors, accoustic links, and other functions of the capsule, the power for the detector is negligible. The power for one year's operation should be obtainable from a single 2 amp-hour storage battery. The power used to run the crystal oscillator should be well regulated and be run from a separate mercury battery for voltage stability. All of the other detector circuits can be supplied from lead-acid cells due to the excellent voltage compatibility of the COS/MOS line.

If power drain were not so important, more than a two to one reduction in package count could be effected by rebuilding the memory out of conventional MOS registers. The power required would increase almost ten-fold. Future developments in the COS/MOS line toward LSI might even reduce the size as much with no sacrifice in power.

TABLE IV. POWER REQUIREMENTS

LOGIC CURRENT	STATIC POWER Microwatts	NUMBER
CD4000 Dual 3-Input plus Inverter	0.01	2
CD4001 Quad 4-Input NOR Gate	0.01	6
CD4002 Dual 4-Input NOR Gate	0.01	2
CD4004 Seven Stage Ripple Counter	1.5	19
CD4006 18 Stage Static Register	0.05	7
CD4009 Hex Inverter	0.05	12
CD4011 Quad 2-Input NAND Gate	0.005	31
CD4012 Dual 4-Input NAND Gate	0.005	3
CD4013 Dual D Flip-flop	0.025	9
CD4014 8 Bit Parrallel in/Serial out	4.0	10
CD4015 Dual 4-Stage Parallel out	4.0	11
CD4016 Quad Bilateral Switch	0.1	6
CD4022 Divide by 8 Counter/Decoder	1.5	2
CD4023 Triple 3-Input NAND Gate	0.005	19
CD4025 Triple 3-Input NOR Gate	0.01	2
TA5872 Dual J K Flip-flop	0.025	4
TA5963 Triple Serial Adder	1.5	7
TA5989 64-Bit Static Shift Register	1.0	151

TOTAL STATIC POWER		268.8	Microwatts
TRANSIENT POWER:	CLOCK	270	
	MEMORY	110	
TOTAL TRANSIENT POWER		380	Microwatts

IV. COMPUTER SIMULATION

A preliminary step in the design of the system components was the evaluation of the required functions through computer simulation. The general algorithm has previously been analyzed [Refs. 1,5]. System evaluation of the actual logic system should be as close to actual hardware circuitry as possible with fixed point numbers and identical computer logic simulation.

A. COMCOR CI 5000

For actual checkout of some of the functions in hardware, the digital patching section of the CI 5000 which is located in the Electrical Engineering Computer Lab, was used. The complementor for the two's complement arithmetic was the first function wired. The second logic block that was tested extensively in hardware before design was finalized was the multiplying scheme. Since a single multiplier used all 32 of the computer's general purpose flip-flops and a major portion of the NAND gates, this method was applicable to only the system blocks; another computer had to be chosen for the overall system evaluation.

B. CDC 160

With the consideration that the simulation should retain as much simularity to the actual hardware as possible, the CDC 160 computer, also located in the Electrical

Engineering Computer Lab, was chosen. The distinct disadvantage, later to become obvious, was the twelve bit word limitation in trying to check out a sixteen bit system. In most areas of the program and logic flow, there were no differences between the designed sixteen bit system and the twelve bit simulation; however, the checkout of the multipliers required further investigation. At this point the 168 arithmatic unit was used as peripheral equipment to allow for double precision.

1. Adaptation to Fixed Point

A magnetic tape was available from the I.G.P.P. containing two records of pressure variations at the 2000 fathom level in an area near San Clemente. The data had been taken at four minute intervals over a four day period with both a Vibratron and a Hewlet-Packard sensor. The data was linearly packed to simulate the required one minute sampling rate, and this prepared tape was used [Ref. 5]. A major portion of the program listed in the Appendix was required to read the BCD tape and convert it into pure binary form. Several problems due to word length limitations were solved by reading in only the four least significant decimal digits and converting them to binary through a table look-up scheme. Simple testing of the resultant value for consistency removed any ambiguity in operations with the numbers except when a change in the fifth BCD digit occurred as noted on Figures 24 and 27. This method was incorporated into the actual design also because the input counters were allowed

to overflow any number of times. Figures 23 and 26 show a record of the tape data compared to the truncated version used on the 160.

2. Precision Considerations

In order to avoid going to double precision and using the external arithmetic device, which would no longer bear any relation to the hardware being simulated, careful study of the data revealed several compromises. In most cases a few of the most significant bits could be dropped because the particular data did not reach the maximum excursion designed for in the sixteen bit system. This, combined with truncating one or two of the less significant bits, reduced the word length to the required twelve bit format. No significant degradation in overall performance was noted from these compromises except in the case of the multipliers. Here, twelve bit precision allows for an error of .002 which is not insignificant since the least-count resolution of the input sensors is on the order of .0005. The multipliers were tested for proper functioning in twelve point individually, but, to allow for system evaluation and to obtain usable results, double precision was used.

The only other notable difference between the simulation and the design was the use of one's as opposed to two's complement arithmetic. Although one's complement requires the additional operation of end-around carries to correct the sum on subtraction, this is handled in the machine; therefore, on the software level, compensations were

made so that the simulation was the same as the two's complement on the hardware level.

3. Simulation Results

The program operated on the input data samples in the same format as the proposed system, sequentially taking one sample value at a time and carrying it through to increment the final integrator storage. Because of the limitation of only 4K of storage in the 160, the input data had to be buffered into the routine 120 values at a time and the output unloaded at 1500 values per run. This incremental method of printing output on the Cal-Comp plotter adds an extra line at each beginning point.

Since the Hewlet-Packard sensor had a very smooth output compared to the Vibratron, it was used for all preliminary runs to test the tide cancellation of the filter section. Once smooth output was obtained from the filter by adjusting the precision and values of the weighting constants, the square and integrate routines were perfected on the Vibratron data. Whether the perturbations detected on the Vibratron tape were noise or actual events is of no concern to the detector's operation but is a problem of sensor design.

Tracing the three interesting "events" detected on the Vibratron tape back to the original data, indicates pressure perturbations on the order of 4mm. Considering the tidal variations as the major noise source of the input data, and these 4mm. variations as signal, the overall improvement

in signal-to-noise ratio is on the order of $2.3 \cdot 10^3$ or 67 db. A more detailed software analysis of system performance, detection thresholds, power spectral densities, and evaluation is included in the Refs. 1 and 5.

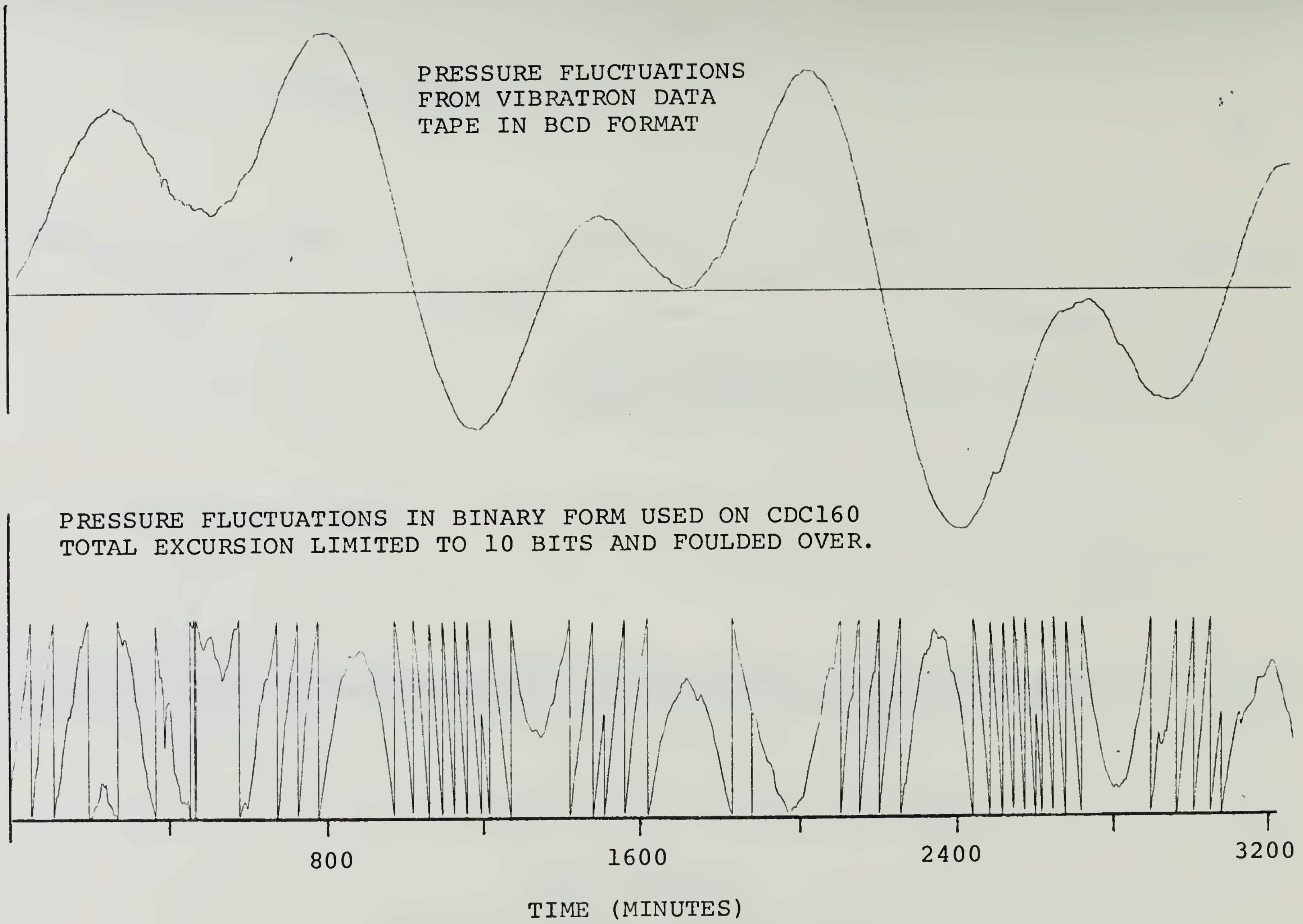


Figure 23. Vibratron Sensor Output

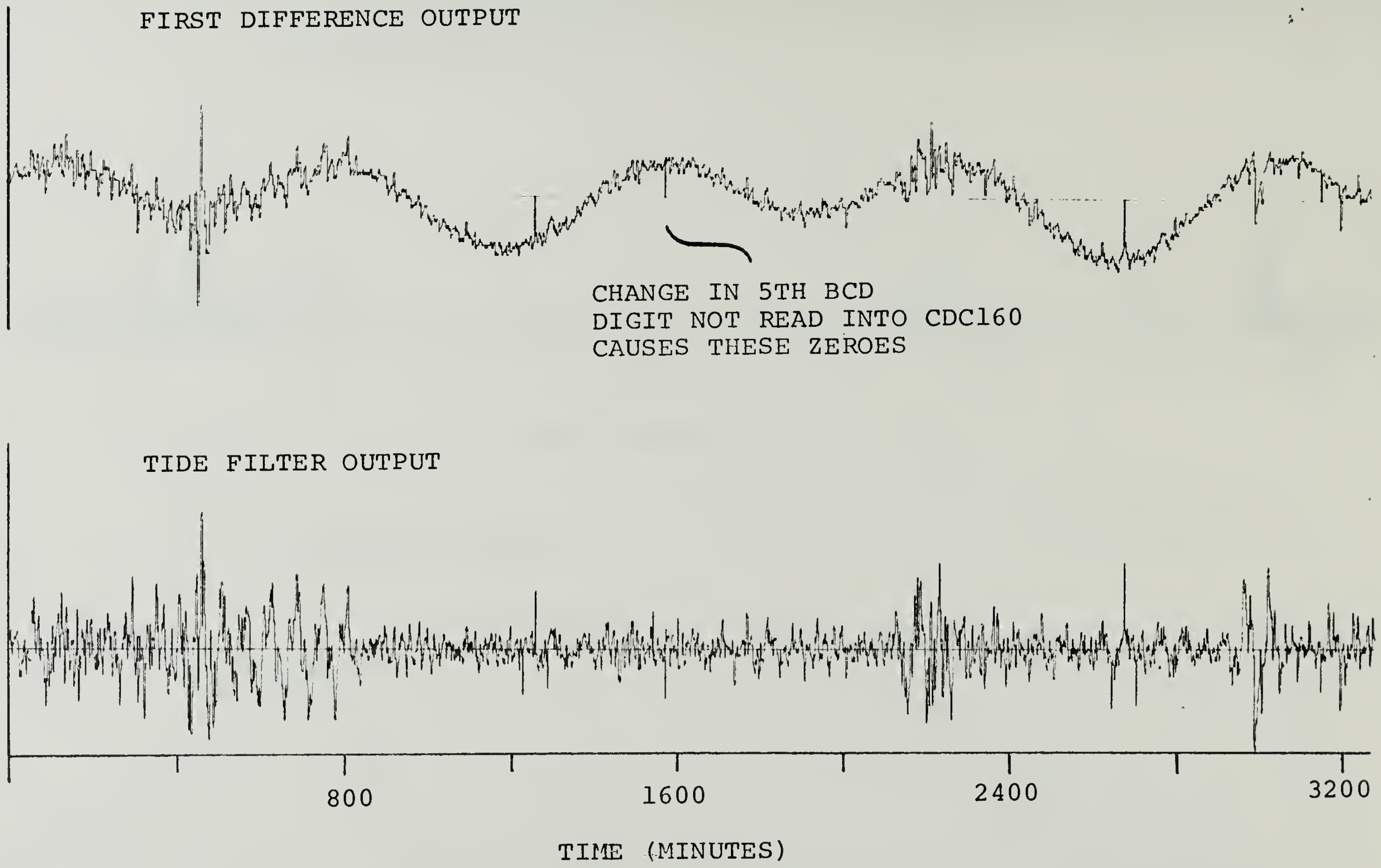
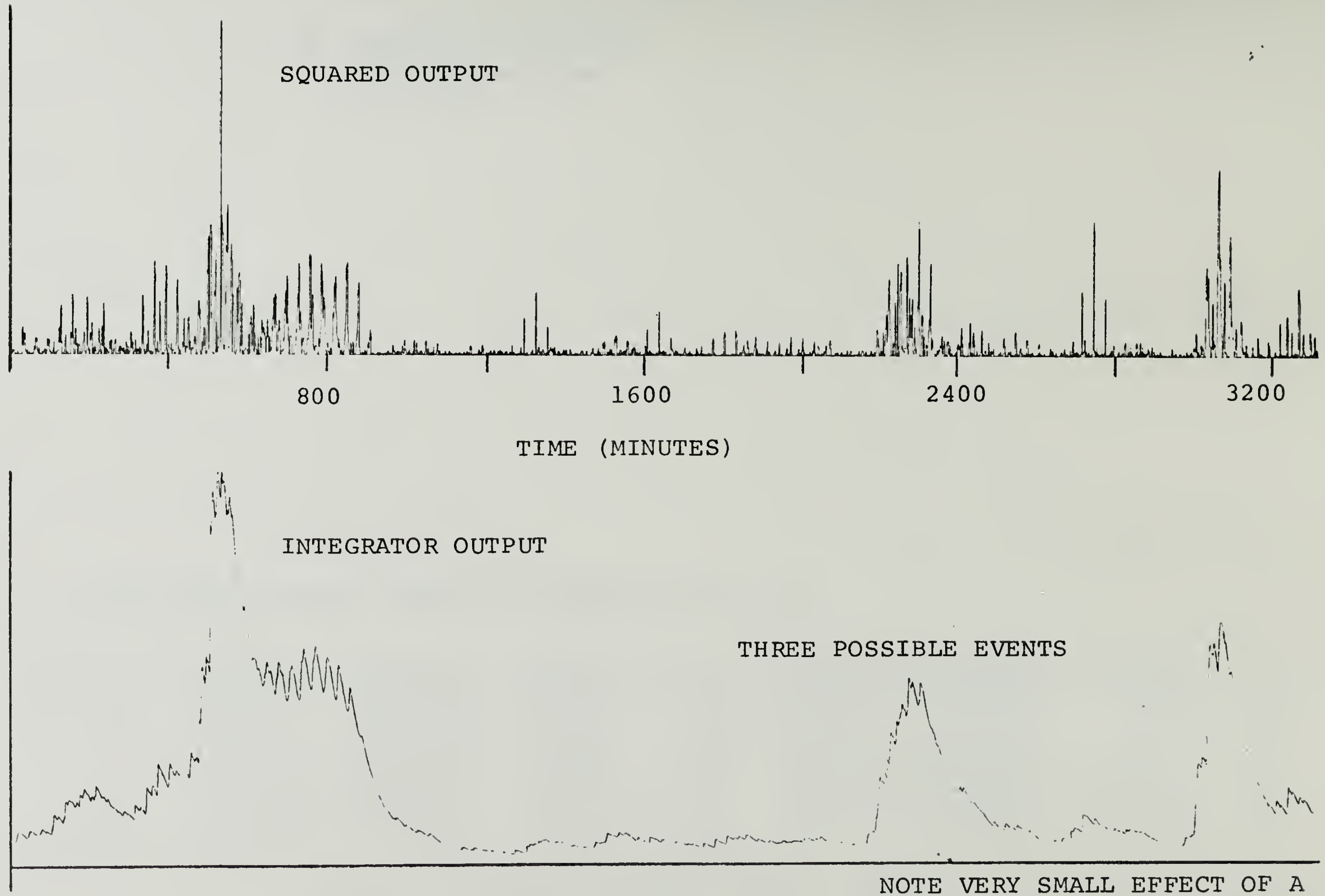


Figure 24. Vibratron Sensor Data



NOTE VERY SMALL EFFECT OF A SINGLE BAD FIRST DIFFERENCE

Figure 25. Vibratron Sensor Data

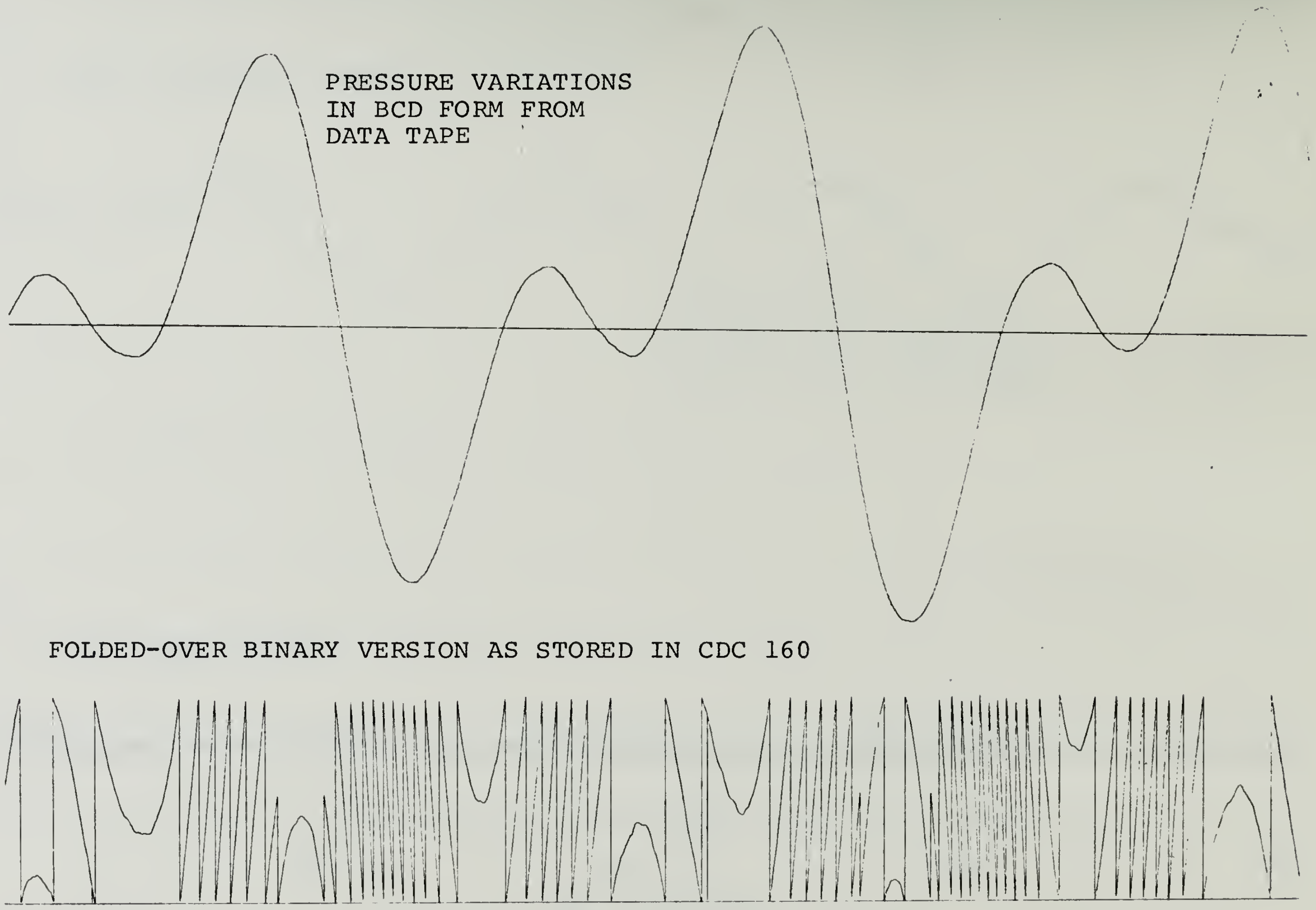


Figure 26. Hewlet Packard Sensor Output

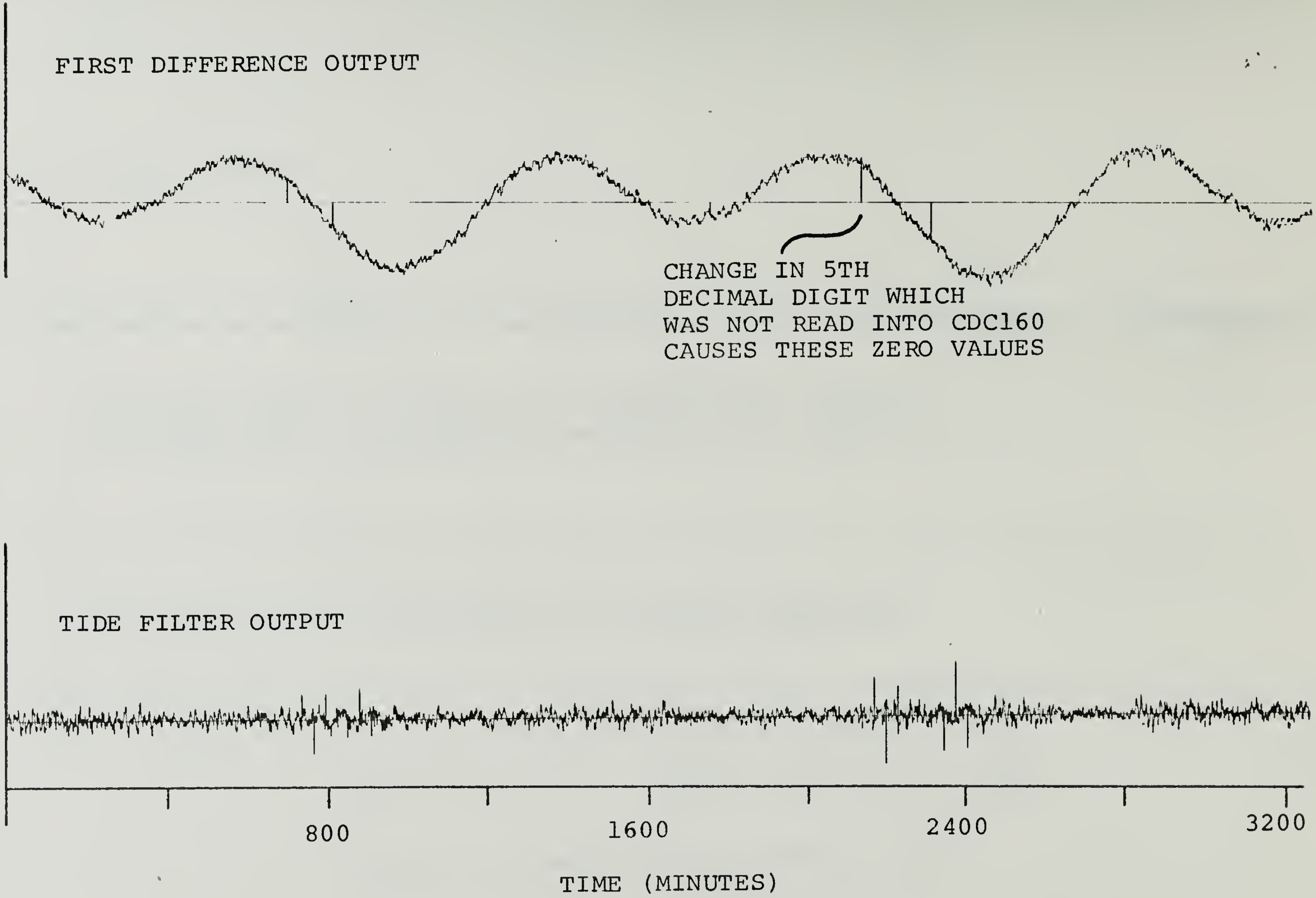


Figure 27. Hewlet Packard Sensor Data

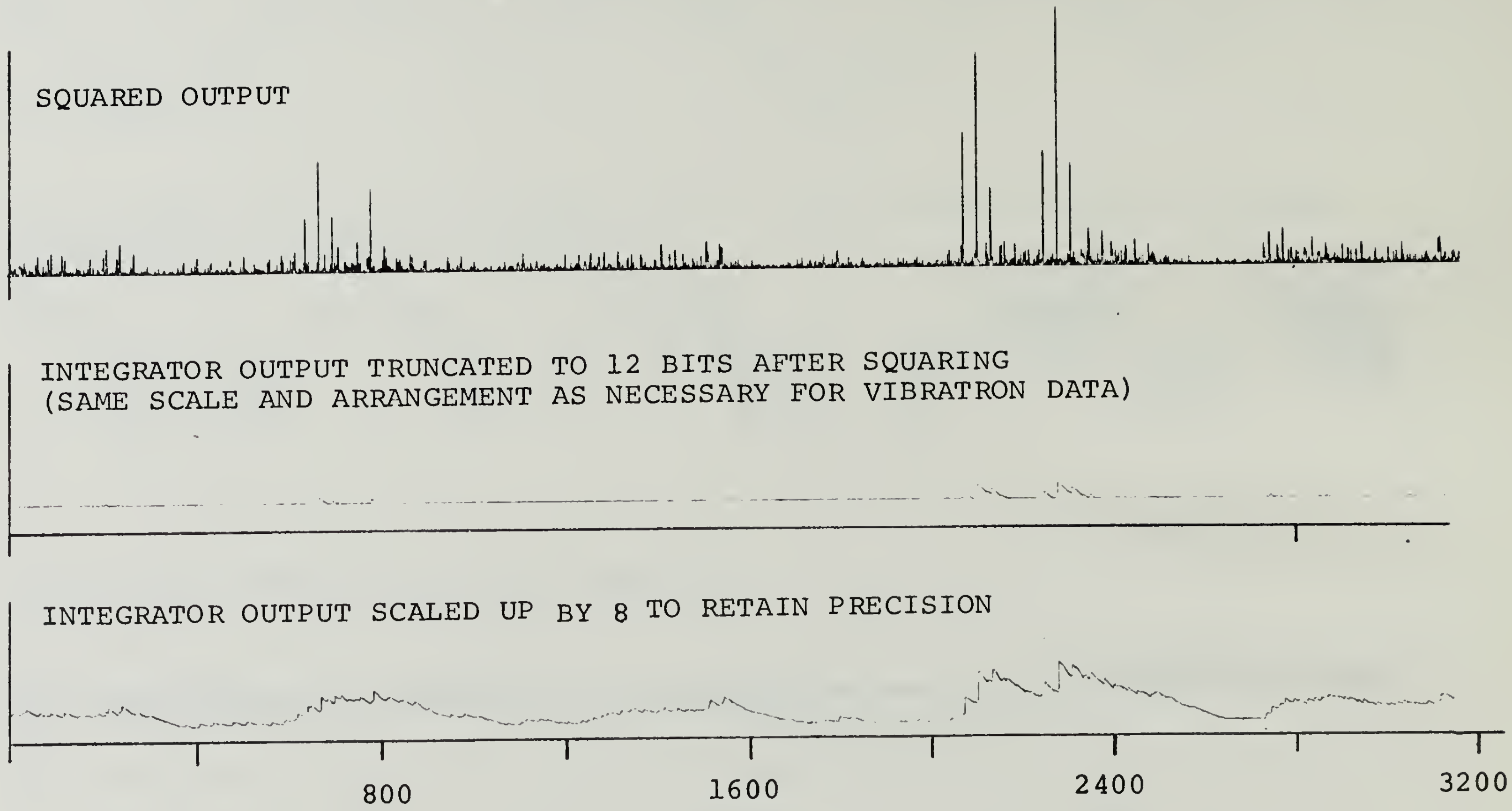


Figure 28. Hewlet Packard Sensor Data

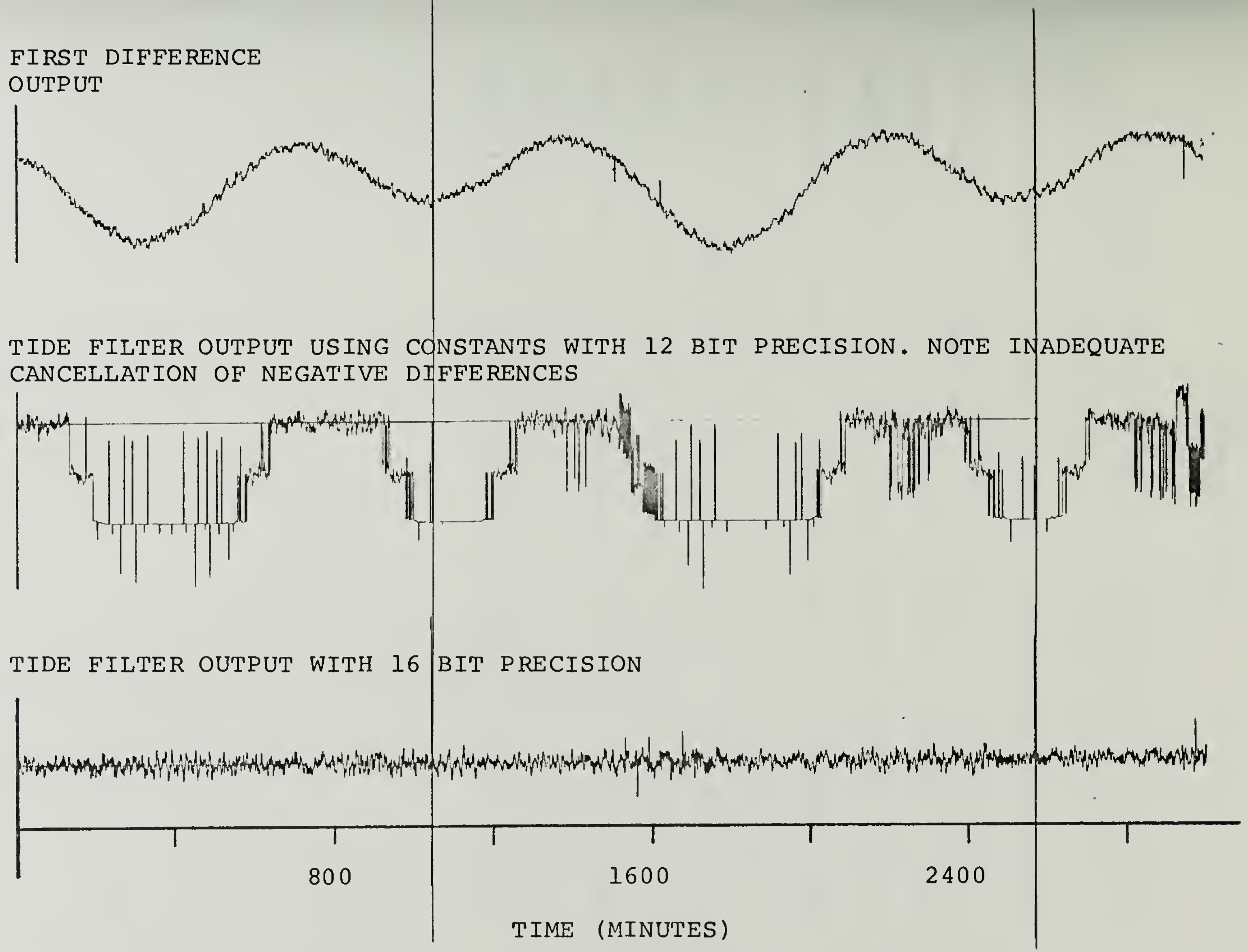


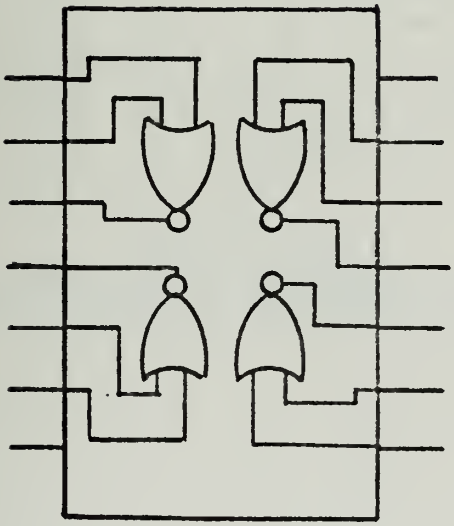
Figure 29. Difference of Precision in Multipliers

V. CONCLUSIONS

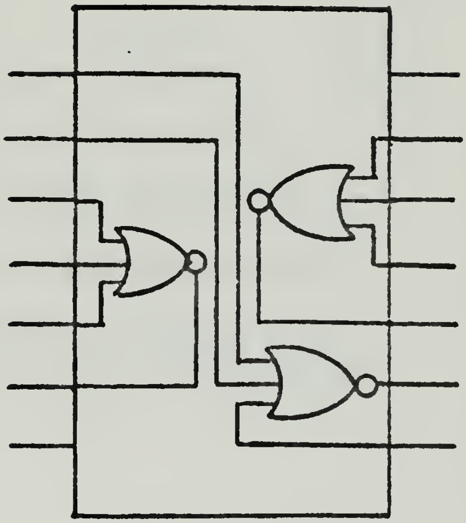
The unusual event detector implemented with COS/MOS logic should provide a very useful and needed package to the growing number of tools used in oceanographic research. The detector can be used as an independent component in a larger system, or, due to its system structure design it can be used as the basis for a rather complex package. Several subsystems for ascent and descent control, telemetry, data storage, and sensor stabilization are actually needed to complete an underwater instrument capsule. The clock and timing circuits of the detector could be easily extended to cover other operations and simple gating of the memory package and addition of more memory cells, could make it a general buffer memory for the other functions attached to the system.

APPENDIX A. COS/MOS LOGIC PACKAGES

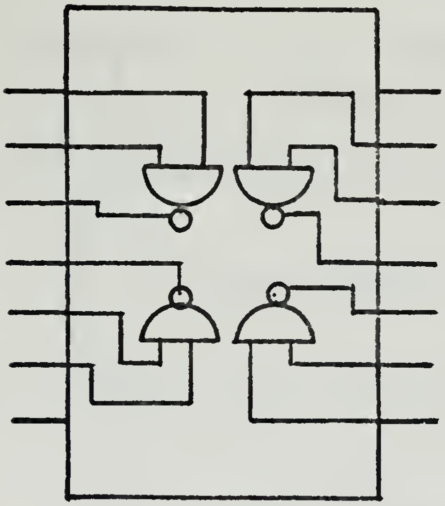
Lead numbers are not given since package styles vary.



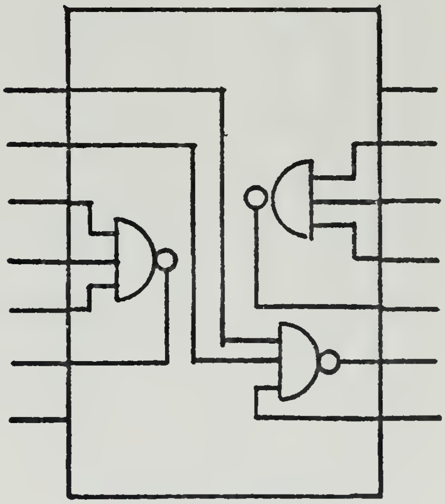
CD4001 QUAD
TWO INPUT NOR GATES



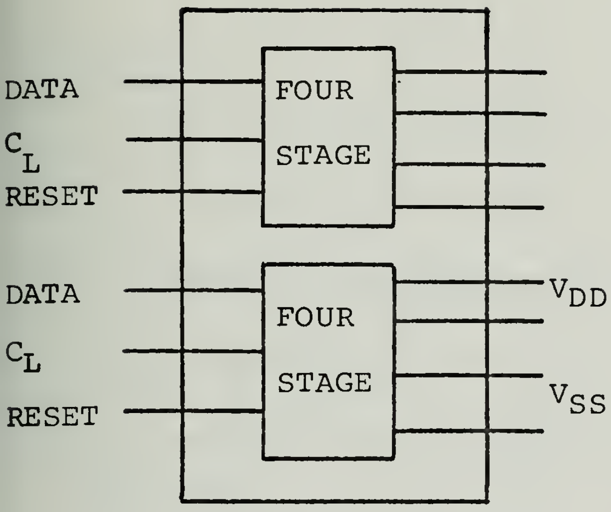
CD4025 TRIPLE
THREE INPUT NOR GATE



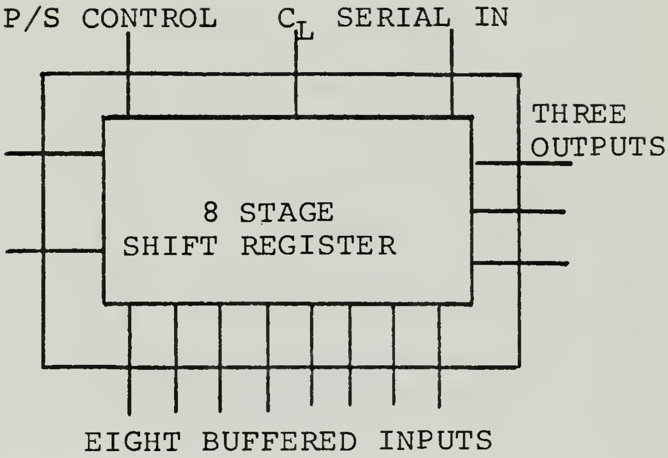
CD4011 QUAD
TWO INPUT NAND GATES



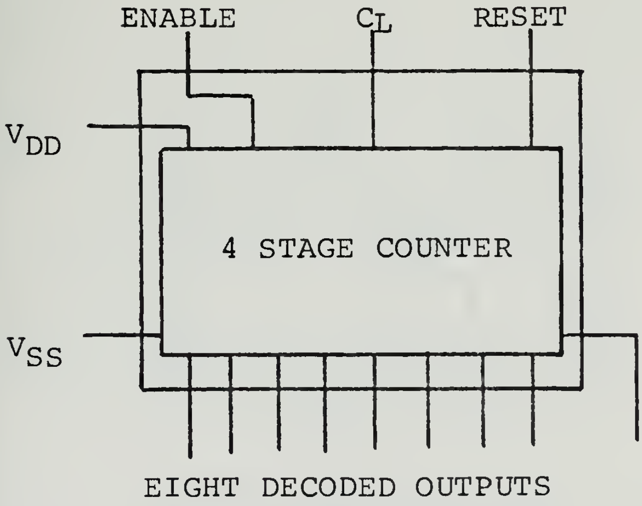
CD4023 TRIPLE
THREE INPUT NAND GATE



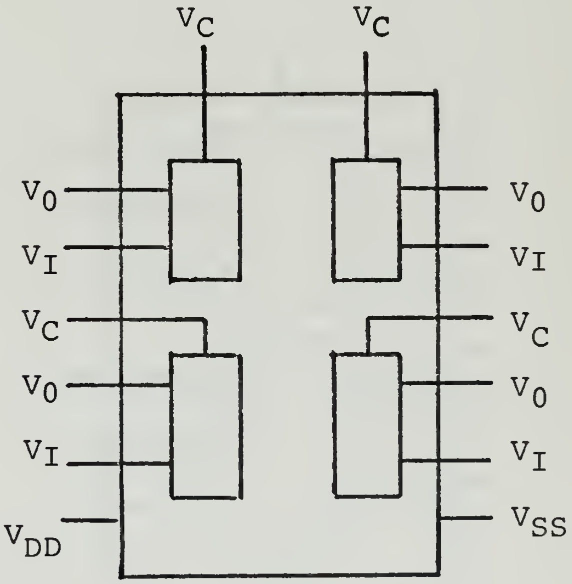
CD4015A DUAL
FOUR STAGE REGISTER
WITH SERIAL IN PARALLEL OUT



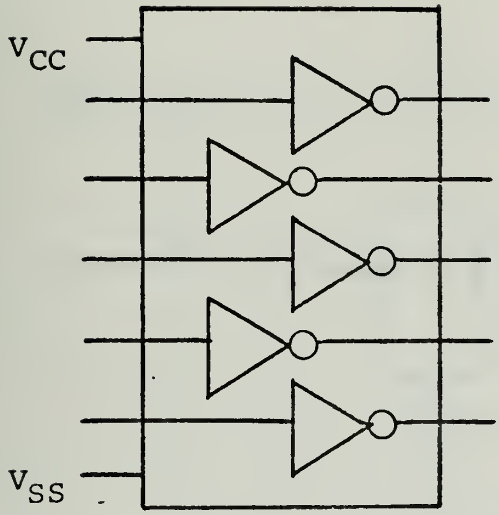
CD4021A 8 STAGE SYNC
PARALLEL INPUT/SERIAL OUT



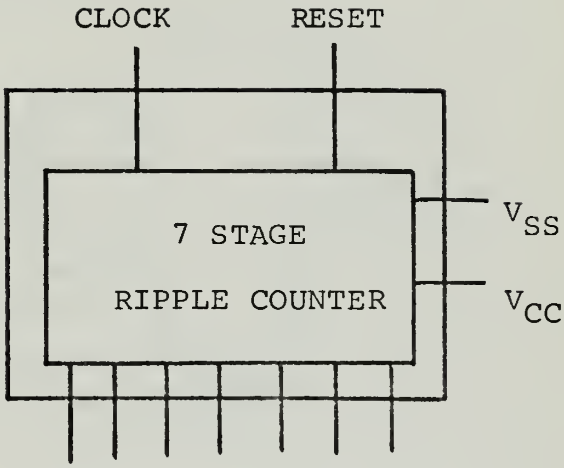
CD4022 DIVIDE BY EIGHT
DIVIDER DECODER



CD4016 QUAD
BILATERAL SWITCH

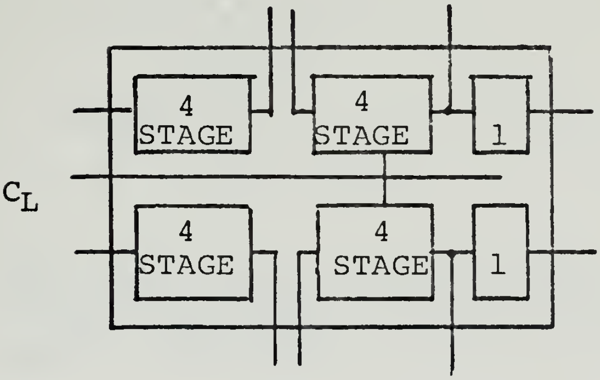


CD4009 HEX INVERTERS

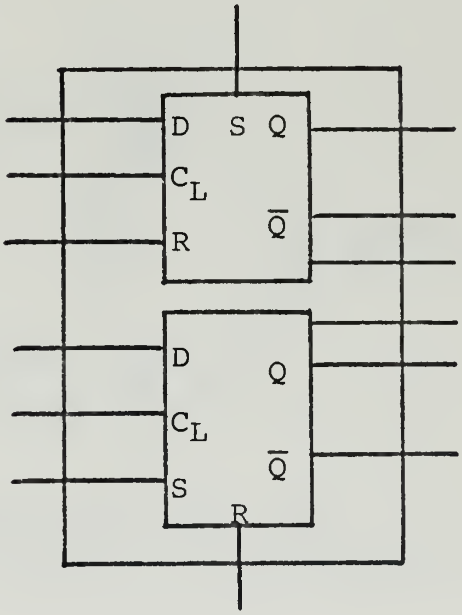


SEVEN BUFFERED OUTPUTS

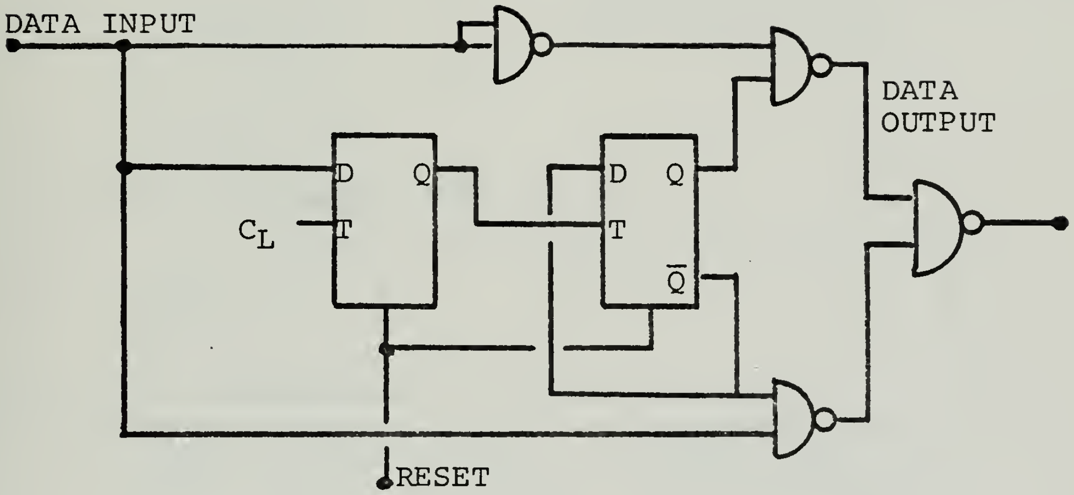
CD4009A 7 STAGE
RIPPLE BINARY COUNTER



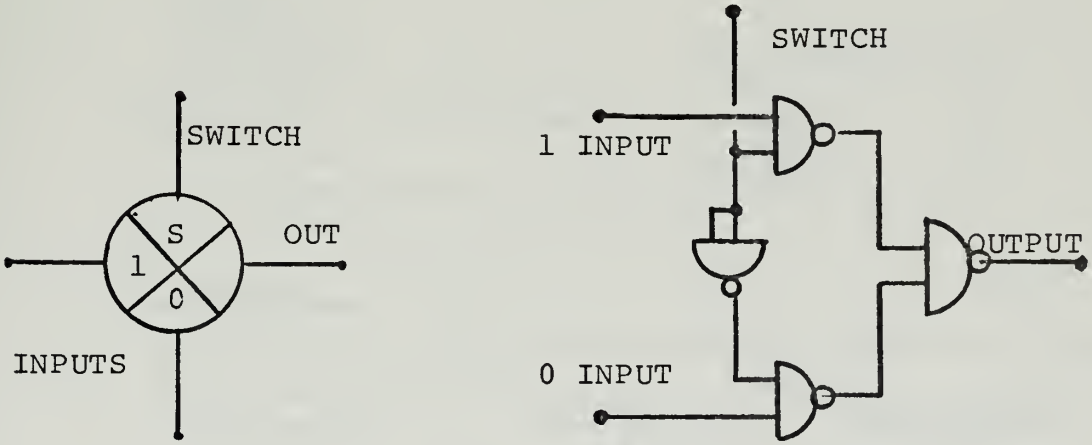
CD4006 EIGHTEEN STAGE
STATIC SHIFT REGISTER



CD4013 DUAL D
FLIP-FLOP

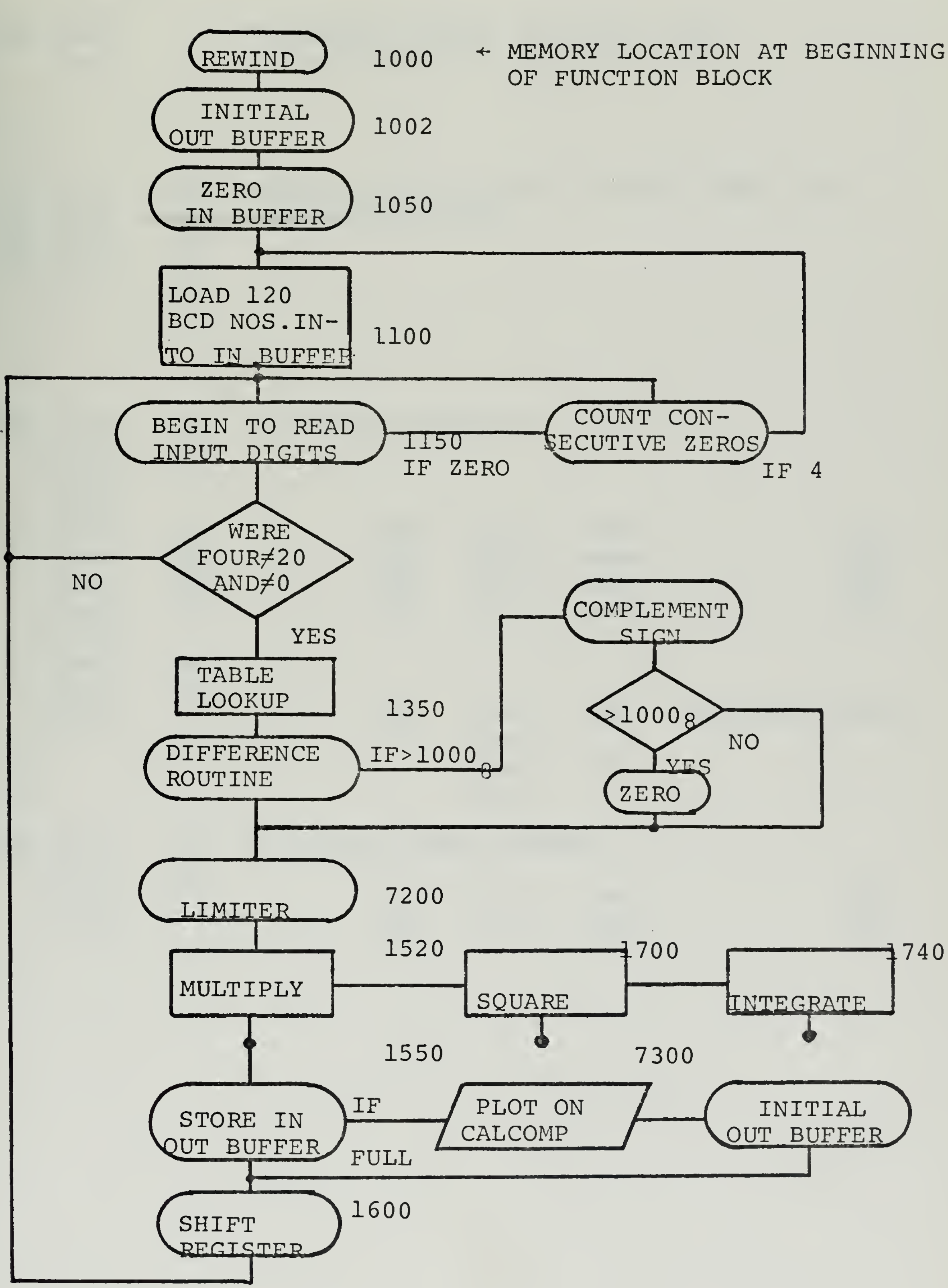


THE COMPLEMENTOR CIRCUIT USED SEVERAL TIMES IN DETECTOR DESIGN CAN BE FORMED FROM ONE CD4013 DUAL FLIP-FLOP AND ONE CD4011 QUAD TWO INPUT NAND GATE



THE SERIAL SWITCH CAN BE FORMED FROM ONE CD4011 QUAD 2 INPUT NAND GATE OR ONE-HALF OF A CD4016 QUAD BILATERAL SWITCH PLUS AN INVERTER

COMPUTER PROGRAM FOR SIMULATION OF THE UNUSUAL EVENT DETECTOR



```
0000  0000         OPERATIONAL ROUTINES FOR VARIOUS ENTRY
      0000              AND EXIT-OUTPUT OPTIONS



1066  0300   nop   REMOVE ALL HALTS EXCEPT AT REWIND, OUTPUT, AND
1115  0300   nop   COMPUTATION COMPLETE
1325  0300   nop
1177  0300   nop



7250  2100   ldm   TO READ OUT INPUT DATA
      0400
      7101   jfi
      1560
1250  2167   ldi             1567   1600   enter             1150
      0720   sbn             1242   0704   enter             0705
      6203   zjf             1245   1260   enter             1250
      7101   jfi             1327   1330   enter             7250
      1260
      0400   ldn
      4167   sti
      0300                   7250   2100   ldm     truncated version
                                    0400
                                    7101   jfi
                                    1560

7240  2030   ldd   TO READ OUT FIRST DIFFERENCE
      0110   ls3
      7101   jfi             1412   0570   enter             0400
      1560                   1425   7200   enter             7240
```

GENERAL LISTING OF TSUNAMI DETECTOR PROGRAM

0000	to	0100	TEMPORARY STORAGE LOCATIONS
0101	to	0377	CONVERSION NUMBERS+ BINARY TO OCTAL
0400	to	0770	SHIFT REGISTER STORAGE
1000	to	1047	GENERAL INITIALIZE AREA
1050	to	1070	ZERO INPUT ARRAY AND BEGIN TO READ
1100	to	1122	READ 120NUMBERS INTO 2000 to 3777
1150	to	1336	BEGIN INPUT CONVERSION TO BINARY
1350	to	1426	DIFFERENCE ROUTINE
1440	to	1513	PREPARE FOR DOUBLE PRECISION
1520	to	1550	CALL MULTIPLY SUBROUTINE
1550	to	1570	SUM THE WEIGHTED DIFFERENCES
1600	to	1637	SHIFT REGISTER IN 0400 to 0570
1700	to	1734	SQUARE ROUTINE
1740	to	1777	INTEGRATE ROUTINE
2000	to	3777	INPUT BUFFER STORAGE
4000	to	6777	OUTPUT BUFFER STORAGE
7000	to	7161	MULTIPLY ROUTINE
7200	to	7224	CLIPPER ROUTINE
7300	to	7375	CAL-COMP PLOTTER ROUTINE
7400	to	7777	RS-022 FLEX AND OCTAL READ ROUTINES

LISTING OF STORAGE LOCATIONS SET ASSIDE FROM 0000 TO 0077

```
0001     into mult for 2 and 4
0002     temp storage of shifts 2
0003     ts of output of mult for 2 and 4
0004     ts value of p entering mult for 2 and 4
0005
0006     ts of shifts 6
0007     ts


0010
0011     ts for shift register routine
0012
0013
0014
0015
0016
0017


0020
0021     0400  location of shift register words one
0022     0401
0023     0600  location of shift register words two
0024     0601
0025
0026
0027


0030
0031     difference 1
0032     difference 31
0033     differrnce 61
0034     difference 91
0035     difference 121
0036
0037


0040
0041     mult diff 1
0042     mult diff 31
0043     mult diff 61
0044   m mult diff 91
0045     mult diff 121
0046
0047
```

0050	
0051	
0052	
0053	
0054	
0055	
0056	
0057	
0060	indigit + correction
0061	running sum first word
0062	running sum 2nd word
0063	address of indigit being operated on
0064	4th digit of converted nos. and how many over 4
0065	7000 store
0066	count of nonzero and not 20 digits
0067	address of first of 4 indigits to be used
0070	reserve all remaining for rs-022
0071	
0072	
0073	
0073	
0074	
0075	
0076	
0077	

```
0000  7101   jfi   LOAD FLEX TAPE USING rso22
      7403




1000  7500   exc   REWIND TAPE AND BEGIN PROBLEM AT START
      1161
      2200   ldc   initialize constants
      4000
      4100   stm
      1561
      4100   stm
      1747


1010  4100   stm
      1771
      0400   ldn
      4000   std
      4001   std   zero the flex read command for safety
      7700   hlt
      7101   jfi
      1050
```

```
1050  2200    ldc   ZERO INPUT ARRAY AND BEGIN TO READ
      2000
      4100    stm
      1056
      0400    ldn
      4100    stm
      2000
      5701    aob


1060  3600    sbc
      3770
      6706    njb
      2200    ldc
      1777
      4063    std
      7700    hlt   halt when array is zero
      7101    jfi
1070  1100
```

```
1150  2200   ldc   BEGIN INPUT CONVERSION TO BINARY
      0100
      4100   stm
      1262
      0400   ldn
      4061   std
      4062   std   zero 61 and 62
      0300


1160  0300
      0300
      0300
1163  5463   aod   step 63 by one and read new digit
      2163   ldi
      6005   zjf   if new digit is zero, step 3 and count
      0400   ldn
      4057   std   zero the zero count
1


1170  7101   jfi
      1220
      5457   aod
      0704   sbn
      6003   zjf   test for 4 consecutive zeroes
      7101   jfi
      1163
      7700   hlt   halt when out of input digits
1


1200  2200   ldc   initialise constants
      2000
      4100   stm
      1122
      4063   std
      0400   ldn
      4057   std
      7101   jfi
      1050         go to zero array and read more indigits
```

```
1220  2163   ldi   BEGIN TO CONVERT TO BINARY
      0720   sbn
      6004   zjf   if 20 step 4
      5466   aod
      7101   jfi   step 66 1 and go read another
      1163
      2066   ldd
      0704   sbn   if it was 20


1230  6205   pjf   are past 4 real nos
      0400   ldn
      4066   std   zero 66
      7101   jfi
      1163         read more
      4064   std
      0400   ldn
      4066   std   zero 66


1240  2063   ldd
      0300
      0704   sbn
      4067   std   store in 67 address of 1st of 4 digits to be converted
      7101   jfi
      1260
```

```
1260  2167  ldi   BEGIN TABLE LOOK UP
      3200  adc
1262  0100
      4060  std
      2061  ldd
      3160  adi
      4061  std   add new digit to running sum
      0300


1270  2200  ldc
      0200
      5060  rad   add 200 to indigit + corr. nos
      2062  ldd
      3160  adi
      4062  std
      0420  ldn
      5100  ram


1300  1262
      3600  sbc
      0200
      6004  zjf   test to see if we have read 4 nos
      5467  aod
      7101  jfi   go back to 1260 and convert more
      1260
      0300


1310  2061  ldd   separate nos into 2 words
      1200  ldc
      7000
      4064  std
      2061  ldd
      3464  sbd   store 12 least significant bits in shift
      4100  stm   register   0400 to 0570
      0400
1

1320  2064  ldd
      0110  ls3
      3062  add
      4100  stm   store most significant bits in 0600 to 0770
      0600
      7700  hlt   halt input to binary is complete, now filter
      7101  jfi
      1330
```

```
1330  2200   ldc   DIFFERENCE ROUTINE
      0400
      4020   std   initialize constants in 20,22,24
      2200   ldc
      0600
      4022   std
      0430   ldn
      4024   std


1340  2020   ldd   begin each difference here
      0601   adn
      4021   std
      2022   ldd
      0601   adn
      4023   std
      2120   ldi   difference word 1, least significant digits
      3521   sbi
1350  4124   sti
      6203   pjf
      7101   jfi
      1372
      2122   ldi   for positive difference of word one
      3523   sbi
      6012   zjf
      6305   njf


1360  0400   ldn   for pos word 1 and pos word 2, zero the difference
      4124   sti
      7101   jfi
      1410
      2124   ldi   for pos word 1 and neg word 2, comlement the sign
      1600   scc
      7000
      4124   sti
1370  7101   jfi
      1410


1372  2122   ldi   for negative difference of word one
      3523   sbi
      6013   zjf
      6206   pjf
      0400   ldn   for neg word 1 and neg word 2, zero the difference
      0300
```

```
1400  4124   sti
      7101   jfi
      1410
      2124   ldi   for neg word 1 andpos word 2, complement the sign
      1600   scc
      7000
      4124   sti
      0300


1410  2020   ldi   test for completion of all 5 differences
      3600   sbc
      0570
      6010   zjf
      0436   ldn   step the difference counter by 36
      5020   rad
      0436   ldn
      5022   rad
1420  0402   ldn
      5024   rad
      7101   jfi   go back to 1340 and calculate another difference
      1340
      7101   jfi   go to spike removal routine
      7200
```

```
1440  0430   ldn   PREPARE FOR DOUBLE PRECISION
      4026   std
      0601   adn
      4027   std
      2126   ldi   for negative nos. store ones in 12 most significant
      6206   zjf   bits
      2200   ldc
      7777
1450  4127   sti
      7101   jfi
      1455
      0400   ldn   for positive nos. zero most significant bits
      4127   sti
      2026   ldd
      0740   sbn
      6206   zjf
1460  2026   ldd
      0602   adn
      4026   std
      7101   jfi
      1442
      2200   ldc
      7000
      4004   std   store multiplication routine access location


1470  2200   ldc   load constant  -3.919921875
      4505
      4042   std
      2200   ldc
      7770
      4043   std
      2200   ldc   load constant  +5.84179700
      2570
1500  4044   std
      2200   ldc
      0013
      4045   std
      2200   ldc   load constant  +1.000000000
      0002
      4025   std
      2200   ldc
1510  0000
      4024   std
      7101   jfi
      1520         jump to call ARITH routine
```

```
1520  0101   pta   CALL MULTIPLY SUBROUTINE
      7004   jpi
      0031         multiply difference 1
      0030
      0024
      0046
      0031         multiply difference 31
      0032
1530  0042
      0050
      0031         multiply difference 61
      0034
      0044
      0052
      0031         multiply difference 91
      0036
1540  0042
      0054
      0031         multiply difference 121
      0040
      0024
      0056
      7101
      1550         jump to summ the weighted differences
```

```
1550  2047   ldd   SUM THE WEIGHTED DIFFERENCES
      3051   add
      3053   add
      3055   add
      3057   add
      0300
      0300
      0300
1560  4100   stm
      4000         store the result in 4000
      5701   aob
      3600   sbc
      6770
      6203   pjf
      7101   jfi   jump to the shift register
      1600
1570  7700   hlt   halt when 4000 to 6770 is full of output
      7101   jfi
      7300
```

•

```
1600  2200    ldc    SHIFT REGISTER 0400 to 0571
      0570
      4011    std
      2011    ldd
      0601    adn
      4012    std
      2111    ldi    shift data to right one sample in 0400 to 0570
      4112    sti


1610  2011    ldd
      0701    sbn
      4011    std
      3600    sbc
      0400
      6612    pjb
      0300
      0300


1620  2200    ldc    shift other half in 0600 to 0770
      0770
      4013    std
      2013    ldd
      0601    adn
      4014    std
      2113    ldi
      4114    sti


1630  2013    ldd
      0701    sbn
      4013    std
      3600    sbc
      0600
      6612    pjb    loop around again
      7101    jfi
      1150           go back to convert another input nos to binary
```

```
1555 4050   std   ENTER THE SQUARE ROUTINE
     7101   jfi
     1700



1700 2200   ldc   square routine
     7000
     4004   std
     2200   ldc   scale factor.....(0400 divides by 2)
     0100
     4054   std
     0300
     0300


1710 0400   ldn   zero most significant digits
     4051   std
     4053   std
     4055   std
     4057   std
     2450   lcd   rectify
     6302   njf
     4050
1

1720 0101   pta   enter ARITH subroutine
     7004   jpi
     0031   multiply   squares input
     0050
     0050
     0052
     0031   multiply   scales output
     0052
1730 0054
     0056
     2057   ldd   load into 56 the squared and shifted results
     7101   jfi
     1560
```

```
1740  0300   nop   INTEGRATION ROUTINE
      2100   ldm
      6770
      4040   std   input first value
      2057   ldd
      1200   lpc
      7770
      0111   ls6
1750  0110   ls3
      4057   std   divide squared input by 8
      2040   ldd
      0111   ls6
      0277   lpn
      4042   std
      2040   ldd
      3442   sbd
1760  6202   pjf
      0400   ldn   takes care of negative zero
      3057   add
      7101   jfi
      1770


1770  4100   stm   store the integrator output in the out. buffer memory
      4000
      5701   aob   increment memory one
      0701   sbn
      4100   stm
      1742
      7101   jfi
      1563         jump back into program


7312  0111   ls6   corrections to CALCOMP PLOTTER ROUTINE
      0110   ls3   for scaling
      0103   ls2
      1200   lpc
      3777
      4011   std
      0300


7365  1771


1007  1771
      2200   ldc   corerections to initial section
      6770
      4100   stm
      1742
      0300
```

```
7300  2200   ldc   CAL-COMP PLOTTER ROUTINE
      4000
      4100   stm
      7311
      7500   exc
      4401
      0400   ldn
      4010


7310  2100   ldm
      4000
      4011   std
      7101   jfi
      7320


7320  2011   ldd   compute difference from last point
      3410   sbd
      4012   std
      6203   pjf
      7101   jfi
      7340
      7101   jfi
      7330
7330


7330  7402   write plus x one
      2012   ldd
      0701   sbn
      4012   std
      6604   pjb
      7401   -x
      7101   jfi
      7350


7340  7401   write minus x one
      5412   aod
      6702   njb
      7101   jfi
      7350
```

```
7350  0300
      7404   write minus y
      5500   aom   step input location one
      7311
      3600   sbc
      6770
      6314   njf
      7700   hlt   halt, output buffer is empty


7360  2200   ldc   initialize storage number 4000
      4000
      4100   stm
      1561
      4100   stm
      1747
      4100   stm
      1771


7370  7101   jfi
      1600
      2011   ldd
      4010   std
      7101   jfi
      7310
```

```
0101  0750          conversion for 1st 3 octal digits of 4th indigit
      0720
      0670
      0640
      0610
      0560
      0530
      0500
      0450
      0000
0121  0144          conversion for 3rd indigit
      0310
      0454
      0620
      0764
      0130
      0274
      0440
      0604
      0000
0141  0012          conversion for 2nd indigit
      0024
      0036
      0050
      0062
      0074
      0106
      0120
      0132
      0000
0161  0001          conversion for 1st indigit
      0002
      0003
      0004
      0005
      0006
      0007
      0010
      0011
      0000
```

```
0301  0001          conversion forlast 2 octal digits of 4th indigit
      0003
      0005
      0007
      0011
      0013
      0015
      0017
      0021
      0000
0321  0000          conversion for last 2 octal digits of 3rd indigit
      0000
      0000
      0000
      0000
      0001
      0001
      0001
      0001
      0000
```

ARITH

A closed subroutine for control of the CDC 168 Arithmetic Unit.

PURPOSE:

This generator routine provides the capability to implement 22-bit fixed point arithmetic sequences using a three address code. Both fractional and integer arithmetic are provided. An indirect addressing option is included.

USAGE:

LOAD the ARITH subroutine anywhere in memory. (Space required 161_8) Store the load address in any convenient low-core directory cell (00XX). CALLING SEQUENCE IS:

```
.
.
.
PTA
JPI     XX                    OP Codes
        OP                    01 - add
         A                    02 - subtract
         B                    30 - multiply fractional
         C                    31 - multiply integer
        OP                    40 - divide fractional
         A                    41 - divide integer
         B
         C                    A and B:  Operands
         .                         C:  Result
         .
         .
(NXT)
```

NOTES:

1. Any number of operations may be called for in sequence prior to return to normal 160 coding.
2. Operands must be in standard format for 22-bit fixed-point arithmetic (1's complement with allowance for sign bit in each word). Least significant half-word in EVEN numbered cell, Most significant half-word in ODD numbered mate.
3. FOR normal direct addressing operands must be referenced only by even numbered address.
4. Use of odd addresses will provide indirect addressing.
5. Cells 70 - 77 are used as temporaries by the program.
6. Timing - Approximately 900 microsecs. per operation.

ARITH SUBROUTINE FOR DOUBLE PRECISION

OPERATIONS USING THE CDC 168

```
                            rem          sr-ARITH
                            rem          control routine for 168 unit
                            rem          mlc - 8/30/64
0100    0602                adn    02
0101    4077                std    77    form address of 1st argument
0102    0470    init        ldn    70    set to fill OP-A-B-C in
0103    4076                std    76       70-71-72-73
0104    0504                lcn    04
0105    4075                std    75
0106    2177    loop        ldi    77    transfer arguments
0107    4176                sti    76    modify as required
                                            if odd(indirect)

0110    0201                lpn    01
0111    6011                zjf    incr
0112    2075                ldd    75
0113    0604                adn    04
0114    6006                zjf    incr
0115    2176                ldi    76
0116    4202                stf    02
0117    2100                ldi


0120    0000
0121    4176                sti    76
0122    5477    incr        aod    77
0123    5476                aod    76
0124    5475                aod    75
0125    6517                nzb    loop
0126    2077                ldd    77
0127    4007                std    07


0130    2200    scode       ldf          set to generate sel code
0131    3323                       3323     3323 for plus
0132    4270                stf    sel      3363 for minus
0133    2070                ldd    70       3303 for mult
0134    0701                sbn    01       3302 for divide
0135    6015                zjf    setout
0136    0701                sbn    01
0137    6011                zjf    sub
```

```
0140   0730           sbn    30
0141   6304           njf    mul
0142   0521   dvd     lcn    21
0143   5257           raf    sel
0144   6106           nzf    setout
0145   0520   mul     lcn    20
0146   5254           raf    sel
0147   6103           nzf    setout


0150   0440   sub     ldn    40
0151   5251           raf    sel
0152   2071   setout  ldd    71      set output addresses A-B
0153   4262           stf    f1
0154   0602           adn    02
0155   4247           stf    t1
0156   2072           ldd    72
0157   4257           stf    f2


0160   0602           adn    02
0161   4245           stf    t2
0162   2070           ldd    70      these addresses OK for
0163   0740           sbn    40      plus, minus, multiply
0164   6335           njf    act
0165   6007           zjf    dvf
0166   0471   dvi     ldn    71      process divide integer
0167   4247           stf    f2


0170   0602           adn    02
0171   4235           stf    t2
0172   6112           nzf    div
0173   6471   hop     zjb    init    HOP links REPEAT to INIT
0174   0471   dvf     ldn    71      process divide fractional
0175   4240           stf    f1
0176   0602           adn    02
0177   4225           stf    t1


0200   2071           ldd    71
0201   4235           stf    f2
0202   0602           adn    02
0203   4223           stf    t2
0204   2072   div     ldd    72      common for divide
0205   4232           stf    f3
0206   0602           adn    02
0207   4224           stf    t3
```

```
0210   2171           ldi     71      sense sign of dividend
0211   6205           pjf     pz
0212   0500   nz      lcn
0213   4071           std     71
0214   4072           std     72      fill neg zero for DD
0215   6104           nzf     act
0216   0400   pz      ldn
0217   4071           std     71

0220   4072           std     72       fill pos zero for DD
0221   7500   act     exf
0222   0000   sel                      activate
0223   7312           out     f1
0224   0000   t1
0225   7311           out     f2
0226   0000   t2
0227   2070           ldd     70

0230   0740           sbn     40
0231   6307           njf     setin   skip unless divide
0232   7305           out     f3
0233   0000   t3
0234   6104           nzf     04
0235   0000   f1
0236   0000   f2
0237   0000   f3

0240   2073   setin   ldd     73      set input address C
0241   4220           stf     f45
0242   0602           adn     02
0243   4203           stf     t4
0244   4207           stf     t5
0245   7214           inp     f45
0246   0000   t4
0247   2070           ldd     70

0250   0730           sbn     30
0251   6103           nzf     03      skip unless MUF
0252   7207           inp     f45
0253   0000   t5
0254   2107           ldi     07
0255   1200           lpf
0256   7700                   7700    sense valid op code
0257   6464   repeat  zjb     hop     no valid - repeat
0260   7007           jpi     07      valid - exit
0261   0000   f45
       0000           end
```

BIBLIOGRAPHY

1. Titus, H.A., Conditional Sampling for Unusual Events, unpublished paper, Naval Postgraduate School, Monterey, California, August 1970.

2. RCA, COS-278, RCA COS/MOS, March 1971.

3. Snodgrass, Frank E., "Deep Sea Instrument Capsule," Science, v. 162, pp. 78-87, 4 October 1968.

4. RCA, "MOS Micropower Complementary Transistor Logic," by M.M. Mitchell and R.W. Ahrons, reprinted from Computer Design, February 1966.

5. Arfman, J.F., A Software Package for Analysis of Geophysical Measurements, M.S. Thesis, Naval Postgraduate School, Monterey, California, June 1971.

6. RCA, Solid State Division, Digital Integrated Circuits Application Note ICAN-6539, Micropower Crystal-Controlled Oscillator Design Using RCA COS/MOS Inverters, January 1971.

7. Caldwell, D.R., Snodgrass, F.E., and Wimbush, M.H., "Sensors in the Deep Sea", Physics Today, pp. 34-42, July 1969.

8. RCA, Solid State Division, Digital Integrated Circuits CD4000A Series, File Number 479, COS/MOS IC's for Low-Voltage (3-15V) Applications, March 1971.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Documentation Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0212 Naval Postgraduate School Monterey, California 93940	2
3.	Assoc Professor H.A. Titus, Code 52 Ts Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	2
4.	Ens. Robert E. Bruninga, USN 1923 Greenbrier Road Florence, Alabama 35630	1

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY *(Corporate author)*

Naval Postgraduate School
Monterey, California 93940

2a. REPORT SECURITY CLASSIFICATION

Unclassified

2b. GROUP

3. REPORT TITLE

Low Power Digital Integrated Circuit Design For an Unusual Event Detector

4. DESCRIPTIVE NOTES *(Type of report and, inclusive dates)*

Master's Thesis;June 1971

5. AUTHOR(S) *(First name, middle initial, last name)*

Robert E. Bruninga

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
June 1971	105	8

8a. CONTRACT OR GRANT NO.

b. PROJECT NO.

c.

d.

9a. ORIGINATOR'S REPORT NUMBER(S)

9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)*

10. DISTRIBUTION STATEMENT

Approved for public release; distribution unlimited.

11. SUPPLEMENTARY NOTES

12. SPONSORING MILITARY ACTIVITY

Naval Postgraduate School
Monterey, California 93940

13. ABSTRACT

A deep ocean capsule is proposed capable of sensing pressure, temperature, flow and direction for a study of tides and internal waves of the oceans. One of the major requirements for this capsule is the detection and recording of unusual events. To meet this need a very low power, highly reliable digital detector has been designed. COS/MOS logic with maximum use of MSI and LSI was used in the design. The detector was simulated using proposed word lengths and fixed point arithmetic which would be utilized in the ensuing hardware. The fixed point simulation was necessary to verify the problems in truncation and round-off errors and to provide necessary design criteria to ensure proper filter functioning and integral detection.

14 KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Deep sea pressure measurements						
Conditional sampling						
Complementary symmetry MOS circuitry						

Thesis
B82637 Bruninga
c.1
133052
Low power digital integrated circuit design for an unusual event detector.